NASA CR-166,123

NASA Contractor Report 166123

NASA-CR-166123
19860005850

PIFCGT - A PIF Autopilot Design Program for General Aviation Aircraft

LIBRARY COPY

JAN 3 1984

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA

John R. Broussard

Contract NAS1-16303
December 1983

~~FOR EARLY DOMESTIC DISSEMINATION~~

~~Because of its significant early commercial potential, this information, which has been developed under a U.S. Government program, is being disseminated within the United States in advance of general publication. This information may be duplicated and used by the recipient with the express limitation that it not be published. Release of this information to other domestic parties by the recipient shall be made subject to these limitations.~~

~~Foreign release may be made only with prior NASA approval and appropriate export licenses. This legend shall be marked on any reproduction of this information in whole or in part.~~

Review for general release December 31, 1985

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

NF02452

FOREWORD

The computer program described in this report was developed by Information & Control Systems, Incorporated (ICS) during the period from November 1980 to August 1981, under Contract NAS1-16303 for the National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia. This work was sponsored by the Avionics Technology Research Branch of the Flight Electronics Division. Dr. David R. Downing and Mr. Wayne H. Bryant served as Technical Representatives monitoring this contract.

The author wishes to express appreciation to Dr. E. Armstrong of the Langley Research Center, for providing ICS with information on the ORACLS program. The author was aided in computer software development by D. Taylor and M. Aherron of the ICS staff.

N86-15320#
~~X81-10069#~~

This Page Intentionally Left Blank

TABLE OF CONTENTS

		page
FOREWORD.		i
I.	PIFCGT OVERVIEW	1
	A. PROGRAM IMPLEMENTATION.	2
	B. DOCUMENTATION	3
II.	PIFCGT DESCRIPTION.	4
	A. PIFCGT CHAPTERS	5
	B. DEFINITION OF VARIABLES	11
III.	SUBROUTINE DOCUMENTATION.	18
	A. BLOCK DATA (AEROD).	19
	B. SAMPLED-DATA REGULATOR (ALPHA AND BETA) . . .	20
	C. SOLUTION OF A*Y*B-X=C (AXBMXC).	22
	D. CREATE WEIGHTING MATRICES (CFWM).	25
	E. FEEDFORWARD MATRICES (CGTPIF)	28
	F. CLOSED-LOOP d MODEL (CLDMOD).	35
	G. CLOSED-LOOP Y MODEL (CLYMOD).	37
	H. CLOSED-LOOP ψ AND Z MODEL (CLXMOD).	39
	I. COMBINE MATRICES (CORNER)	41
	J. DIAGONAL PARTITION (DIAGPAR).	43
	K. FORM A DIAGONAL MATRIX (DIAGPUT).	45
	L. DIMENSIONS (DIMSS).	47
	M. DISCRETE MODEL CONSTRUCTION (DISCMOD)	48

TABLE OF CONTENTS (CONTINUED)

		page
N.	DISCRETE PIF CONTROL LAW SIMULATION (DPIFS) .	51
O.	DISCRETE RICCATI EQUATION SOLUTION (DREG) . .	56
P.	SUBSET OF A MATRIX (EXTR)	60
Q.	F AND G CONSTRUCTION FROM AERODYNAMIC COEFFICIENTS (FGAERO)	62
R.	EIGENVALUES/EIGENVECTORS OF CLOSED-LOOP PLANTS USING CSQZ (FREEF)	70
S.	EIGENVALUES/EIGENVECTORS OF CLOSED-LOOP PLANTS USING EIGEN (FREEO).	74
T.	ACCELERATION AND VELOCITY VECTOR OBSERVATION MATRIX CONSTRUCTION (HADAC)	77
U.	FM, GM, HM, DM, H AND D MATRIX CONSTRUCTION (HDCON)	83
V.	INTERPOLATE AERODYNAMIC DATA (INTERP)	89
W.	MULTIPLY A TRANSPOSE TIMES B TIMES A (MATBA).	90
X.	REARRANGE MATRIX (MOVEALL).	92
Y.	PIF EIGENVALUE PLACEMENT (PIFEIG)	94
Z.	PIF CLOSED-LOOP EIGENVALUE EIGENVECTOR CALCULATIONS (PIFFRE)	99
AA.	PIF GAIN COMPUTATIONS (PIFG).	101
BB.	CONSTRUCT THE PIF MODELS (PIFMODL).	106
CC.	PIF EIGENVALUE PLACEMENT (PIFPLC)	110
DD.	OUTPUT NUMERICAL DATA (PRNT).	114
EE.	PRINT TITLE (PRNTITL)	116
FF.	OUTPUT LABELED NUMERICAL DATA (PRNT2)	118

TABLE OF CONTENTS (CONTINUED)

page

GG. SAMPLED-DATA REGULATOR USING ALPHA AND BETA (QRMHAT). 121

HH. SAMPLED-DATA REGULATOR USING ORACLS (QRMP1) . 126

II. REDUCE MATRICES (REMAT) 131

JJ. STEP RESPONSE EIGENVALUE INFORMATION (RESPON). 134

KK. PRINT OUT INFORMATION (RUINFO). 136

LL. CREATE A (3XN) MATRIX FROM 3 DIFFERENT 1XN MATRICES (R3MAT). 137

MM. SCHUR FORM SOLUTION OF A * X * B - X = C (SHRSOL). 139

NN. ROUND OFF SMALL NUMBERS (SMALL) 141

OO. ITERATIVE REFINEMENT SOLUTION OF A * X * B - X = C (SOLVER). 142

IV. SUMMARY - DESIGN PROCEDURE 145

REFERENCES. 147

APPENDIX A. 148

APPENDIX B. 167

APPENDIX C. 169

I. PIFCGT OVERVIEW

The PIFCGT program is a collection of FORTRAN-coded subroutines which can be used to formulate, design, and evaluate a proportional-integral-filter (PIF) digital flight control law. The PIF control law is designed using the Linear Quadratic Regulator (LQR) formulation to obtain feedback gains and command generator tracker (CGT) theory to obtain feedforward gains. Structurally, the PIFCGT program is driven by a main executive which calls individual modular subroutines. The PIFCGT main executive is written using top-down structure and depends on user input to determine which of the program subroutines are to be used for a given control law design.

Flexibility in the PIFCGT program is accomplished by using a general multivariable control design approach and employing dynamic (vector) data storage. In almost all cases, data arrays in PIFCGT subroutines are treated as packed one-dimensional arrays which can' easily be passed between subroutines without a maximum array size parameter appearing as an argument of the calling sequence. The dynamic storage capability allows subroutine program size to be specified and controlled through the PIFCGT main executive. The use of common blocks has been reduced to a minimum (PIFCGT, AEROD, CRWM, DPIFS, FGAERO, RUNINFO) in order to increase subroutine modularity and portability.

The PIFCGT program is explicitly written for a specific application: the design of any one of nine PIF autopilots for the NAVION general aviation aircraft. The PIFCGT program has been developed as part of the effort in NASA Langley Research Center's General Aviation Terminal Area Operation Research (GATOR) program. The GATOR program is directed at developing and evaluating advanced flight control, guidance and display concepts that make use of the recent advances in digital flight control hardware, digital control theory, navigation and landing systems, and electronic displays.

This report constitutes the second part of a two part report. The first part in Ref. 1 is a discussion of the derivation, design, implementation, and flight testing of the PIF control law autopilots. This report documents the software that is used by a PIF control law designer to produce PIF control gains, evaluate PIF closed-loop performance, and determine PIF linear simulations to command inputs.

A. PROGRAM IMPLEMENTATION

The PIFCGT program was developed using the facilities of the NASA Langley Research Center Computer Complex. The program was written for execution on a CDC 6600 using the FORTRAN FTN4 compiler.

PIFCGT requires $140{,}000_8$ octal locations for execution using a segmentation procedure. PIFCGT requres $224{,}300_8$ octal locations for execution without segmentation. The maximum number of model states allowed is 20 (12 aircraft states + 4 control states + 4 integrator states in a fully coupled PIF aircraft control design). The maximum number of control and command states allowed is 4.

A command file consisting of the JOB control information is required to execute each of the autopilot designs under the NASA Langley NOS 1.3 operating system. A command file example for the NOS 1.3 operating system is presented in Appendix B along with the segmentation directives in Appendix C.

All of the data needed by PIFCGT from a user designing one of the nine autopilot is read from a NAMELIST file. The NAMELIST files used in the NAVION PIF designs for each autopilot are presented in Appendix A.

The PIFCGT program can be easily extended to design one of the nine autopilots for a variety of aircrafts other than the NAVION by simply changing the aerodynamic data base in the BLOCK DATA (AEROD) subroutine. Changes to the aerodynamic data base can also be performed using the NAMELIST variables. A new autopilot command system (autothrottle for example) not currently available in PIFCGT can be incorporated into the program by adding (i.e., not modifying) software in three subroutines (HDCON, DPIFS, and PIFG) in the form of alternative "GO TO XX" code sections. This built-in "expandability" feature is a special feature of PIFCGT that makes new autopilot designs straightforward.

B. DOCUMENTATION

The next two chapters in the report are directed to input and output requirements of the programs and subroutines. Chapter 2 is devoted to the main program PIFCGT. A step by step discussion is given for the flow of the program and all variables used in the program are defined in a Table. Chapter 3 describes each of the primary level subroutines used by PIFCGT. The secondary level subroutines are composed of subroutines from the ORACLS library system (ORACLIB), Ref. 2, and specialized subroutines written by Information & Control Systems. Documentation for ORACLIB can be found in Ref. 2 and is not presented in this report.

II. PIFCGT DESCRIPTION

PIFCGT uses a top-down structure that depends on user input to determine which program chapters are to be used for a given design. There are five main chapters: data initialization and set up, PIF matrix construction, feedforward matrix construction, PIF design and performance analysis, and PIF time history computation. Two flags control the main logic, ICH(I,J) and ICHOSE. ICH(I,J) indicates which chapters are to be executed; I represents the ith chapter and J represents the jth section within the ith chapter. ICHOSE is an integer which indicates which autopilot mode is to be designed by the program. The nine autopilot ICHOSE options are shown in Table 1. Two of the APF LOC autopilots may eventually be discontinued depending on flight test results.

During the course of developing PIFCGT, print statements have been used to check intermediate calculations. All the print statements remain in PIFCGT, but are inactivated by setting logical variables beginning with the 4 letters "DEBU" to FALSE. Table 1 lists the areas of software for which each "DEBUG" variable activates print statements.

Two features may be added to the PIFCGT program by a user to enhance its capabilities. The ORACLS subroutines, EIGEN (used by FREEO, Section S) sometimes has numerical difficulties computing eigenvalues with sufficient accuracy. An alternative eigenvalue/eigenvector procedure not available in ORACLS (but available in FTNMLIB - a NASA Langley computer center math library) is discussed in FREEF, Section R. The second feature is a plotting capability. A ploting package was developed for PIFCGT but is highly machine dependent and is not documented in this report.

A. PIFCGT CHAPTERS

This section presents a list of the chapters that can be activated by the logical variable ICH. The format for the presentation of the PIFCGT chapters consists of the logical array variable element, the name of the program or subroutine that the logical element activates and a brief description of the function performed. The user should follow the logic in the PIFCGT main executive program while reading this section.

ICH(1,1) (PIFCGT) INITIALIZE CONSTANTS

PIFCGT begins by calling the ORACLS subroutine RDTITL. RDTITL defines data statements in internal ORACLS common blocks and reads a user defined title card which is shown at the end of the submit file in Appendix B.

ICH(1,2) (PIFCGT) READ IN FLAGS AND DESIGN PARAMETERS

The NAMELISTS NAM1, NAM2, NAM3, NAM5, NAM6, AERO1, AERO2, and NAM7 are read in from logical unit 20. If "DEBUXX" variables are TRUE, the NAMELISTS are printed using the namelist output format.

ICH(1,3) (RUNINFO) PRINT OUT FLAGS AND DESIGN PARAMETERS

The subroutine RUNINFO is called for the first time. RUNINFO is the information subroutine that echos the flag conditions and chapter calls the user has specified in the namelists.

ICH(1,4) (INTERP) INTERPOLATES AERODYNAMIC DATA AT VTP

The aerodynamic coefficients in the common blocks AERO and AER1 are assumed to be for two different straight and level trim conditions (a lower and a higher forward velocity, respectively). INTERP uses the variable VTP as an intermediate velocity to interpolate the coefficients for a trim condition at VTP. The interpolated values overwrite the values in AERO.

ICH(1,4) (FGAERO) CONSTRUCT F AND G (BODY AXIS) MODEL MATRICES

The nonlinear equations of motion of the aircraft have been explicitly linearized about a straight and level flight condition. The linearization equations and the aerodynamic data in AERO are used in FGAERO to construct the (12 x 12) plant system matrix, F, and the (12 x 4) control effect matrix, G.

ICH(1,5) (FGAERO) TRANSFORM F AND G SO ANGLES AND CONTROLS ARE IN DEG

FGAERO constructs F and G with units of feet for distance and radians for angles. If ICH(1,5) is TRUE (as is the case in all NAMELISTS in Appendix A), F and G are converted to use angular units of degrees. Units of feet and radians produce numerically ill-conditioned matrices in later computations.

ICS(1,10) (DIMSS) SET UP DIMENSION PARAMETERS

ORACLS requires that each matrix vector have a two-dimensional integer vector that specifies the first dimension of the matrix in the first location of the integer vector and the second dimension of the matrix in the second location. DIMSS initializes the two-dimensional integer vectors.

ICH(1,11) (PIFCGT) PRINT OUT THE F AND G MATRICES

ICH(1,15) (FREEO) FIND EIGENVALUES AND EIGENVECTORS OF OPEN-LOOP F MATRIX

ICH(1,16) (HADAC) CONSTRUCT HA AND DA ACCELERATION OBSERVATION MATRICES

Later in the construction of the PIF control law (PIFG) some of the state feedback elements (Δu, Δv, Δw) can be replaced with accelerometer output feedback (Δa_x, Δa_y, Δa_z) by transforming the feedback gains. The matrices HA and DA mathematically relate the linearized accelerometer output with the body-axis aircraft states and controls using

$$\Delta\underline{a} = HA * \Delta\underline{x} + DA * \Delta\underline{u} \tag{1}$$

and are required to perform the feedback gain transformation.

ICH(2,2) (HDCON) DEPENDING ON ICHOSE CONSTRUCT H, D, FM, GM, HM, AND DM MATRICES

The command output matrices (H and D) and command model matrices (FM, GM, HM, and DM) are constructed in HDCON depending on the value of ICHOSE. Logical vectors, XR, UR, YR, XMR, and UMR are initialized for later use in REMAT.

ICH(1,3) (RUNINFO) PRINT OUT FLAGS AND DESIGN PARAMETERS

RUNINFO is called for the second time and echos values of parameters constructed in subroutines and specified in NAMELISTS.

ICH(2,4) (CFWM) CONSTRUCT COST FUNCTION WEIGHTING MATRICES; A, R, AM, RD, QZ

Linear Quadratic Cost function weighting elements in the common block QWHT are used to form the cost function weighting matrices Q, R, AM, RD, and QZ.

ICH(2,5) (PIFMODL) CONSTRUCT PIF MODEL MATRICES PHI, GAMA, QHAT, RHAT

The continuout-time mathematical model matrices for the PIF Linear Quadratic Regulator problem are formed.

ICH(2,6) (REMAT) REDUCE MATRICES BY ELIMINATION

Using the logical vectors XR, UR, YR, XMR, and UMR, undersired states, controls and commands are eliminated from the PIF matrices. The result is that the fully coupled PIF design matrices are reduced in dimension. The reduction performed in REMAT usually leaves only longitudinal related states or lateral-directional related states.

ICH(2,7) (PIFEIG) FORM DESIRED CLOSED-LOOP MODEL FROM PLANT MODEL

The Riccati matrix equation solution procedure in DREG requires a starting stabilizing PIF gain to find a solution. The ORACLS subroutine, DSTAB, often is not able to numerically find a stabilizing gain matrix. PIFEIG, in combination with PIFPLC, finds a stabilizing gain matrix using an eigenvalue/eigenvector placement procedure. PIFEIG constructs a PIF model that has stable eigenvalues. The procedure works even if the PIF model is stabilizable but not controllable.

ICH(2,8) (PIFMODL) REPLACE U FEEDBACK WITH AX FEEDBACK IN HX DX

ICH(2,9) (PIFMODL) REPLACE V FEEDBACK WITH AY FEEDBACK IN HX DX

ICH(2,10) (PIFMODL) REPLACE W FEEDBACK WITH AZ FEEDBACK IN HX DX

ICH(2,12) (PIFMODL) REPLACE AX FEEDBACK WITH VT FEEDBACK IN HX DX

ICH(2,13) (PIFMODL) REPLACE AY FEEDBACK WITH BETA FEEDBACK IN HX DX

ICH(2,14) (PIFMODL) REPLACE AZ FEEDBACK WITH ALPHA FEEDBACK IN HX DX

The feedback gain in PIFG is transformed to allow for different sensors. AX, AY, and AZ are the body mounted accelerometer sensor observations. VT, BETA, and ALPHA are the pitot tube, sideslip vane and angle of attack vane sensor observations. If ICH(2,12) is TRUE, then ICH(2,8) must be TRUE, etc. HX and DX are the sensor observation matrices. Observation matrices for VT, BETA, and ALPHA are formed in HADAC.

ICH(2,11) (PIFEIG) USE RANDOM NUMBERS TO HELP FIND PHICL

The stable model formed in PIFEIG and stored in PHICL uses the feedback gain GK during construction. The elements in GK are generated internally in PIFEIG if ICH(2,11) is TRUE or must be passed to PIFEIG by the designed from NAM5 if ICH(2,11) is FALSE.

ICH(3,1) (DISCMOD) FORM DISCRETE EQUIVALENT OF CONTINUOUS MODEL AND PLANT

If ICH(3,1) is FALSE the feedforward matrix construction scalar, SSS, is set to 0.0 and DF, DG, DFM, and DGM are set to equal F, G, FM, and GM, respectively. If ICH(3,1) is TRUE the feedforward matrix scalar, SSS is set to 1.0 and the discrete plant and model representations are obtained for a zero-order hold in the control.

One of the objectives of the lateral-directional command model is to cause the sideslip to be zero when the aircraft is banked in a turn. When designing the ICHOSE= 9 (APR LOCI) autopilot, the second element in DGM must be adjusted so that the feedforward matrix element S12(1) is zero. S12(1) * ϕ_c is the steady state side velocity. The second element in DGM is adjusted by the program automatically, a maximum of IKDGM times using the feedback equation

$$DGM(2) = DGM(2) + DKDGM * DELT * S12(1) \tag{2}$$

IKDGM and DKDGM are specified by the designer so that the iterative adjustment procedure shown in Eq. 2 is stable.

If ICHOSE=15 or 16 (APR LOCR or APR LOCP autopilots) the digital model matrix element DGM(3) is set to zero. Note: If ICHOSE=16 is the desired autopilot design, the subroutine HDCON makes the appropriate adjustments, then resets ICHOSE to ICHOSE=15. The ICHSOE=16 (APR LOCP) design is performed this way because it is very similar to the ICHOSE=15 command model (APR LOCR).

ICH(3,2) (CGTPIF) SOLVE FOR FEEDFORWARD MATRICES

A description of this subroutine, which is an involved procedure for finding the feedforward matrices S11, S21, S22, and S23, is given in Chapter 3, Section E.

ICH(4,2) (QRMP1) FORM DISCRETE EQUIVALENT OF CONTINUOUS SYSTEM USING ORACLS SUBROUTINES

ICH(4,3) (QRMHAT) FORM DISCRETE EQUIVALENT OF CONTINUOUS SYSTEM USING ALPHA AND BETA SUBROUTINES

These two subroutines transform a continuous-time Linear Quadratic Regulator PIF control problem into a sampled-data regulator PIF control problem. ORACLS fails to perform the transformation if there is only one control; QRMHAT must be used under this condition. When both subroutines execute properly, their answers are identical.

ICH(4,4) (PIFPLC) PLACE THE EIGENVALUES AND EIGENVECTORS USING PIF CONTROL

PIFPLC uses the discrete PIF plant representation and the discrete plant representation of PHICL (constructed in QRMHAT (or QRMP1)) to determine a stabilizing feedback gain using eigenvalue/eigenvector placement.

ICH(4,5) (PIFPLC) FIND LOG OF F + GK IN PIFPLC BEFORE FINDING EIGENVALUES

After PIFPLC determines the stabilizing feedback gain, the resulting closed-loop eigenvalues are printed out if DEBU44 is TRUE. If ICH(4,5) is TRUE, the discrete-time closed-loop plant is transformed to the equivalent continuous-time plant using the log of a matrix before determining the eigenvalues.

ICH(4,6) (DREG) SOLVE DISCRETE RICCATI EQUATION FOR FEEDBACK GAINS

The discrete-time algebraic Riccati equation is solved using Newton's method. If ICH(2,7) is TRUE, the initial feedback gain GK is assumed to be stabilizing. If ICH(2,7) is FALSE, the subroutine DREG attempts to find an initial stabilizing gain, using the ORACLS subroutine DSTAB. DSTAB is not always successful.

ICH(4,7) (LOGX1) FIND EQUIVALENT CONTINUOUS SYSTEM FOR A CLOSED-LOOP PIF CONTROLLER

The log of a matrix is used to transform PHICL=PHI-GAMA*GK to an equivalent continuous plant. The closed-loop eigenvalues are stable in the left-half complex plane.

ICH(4,8) (PIFFRE) FIND CLOSED-LOOP EIGENVALUES AND EIGENVECTORS

The ORACLS subroutine EIGEN is used to find the eigenvalues and eigenvectors of PHICL. EIGEN is sometimes inaccurate.

ICH(4,9) (PIFG) COMPUTE THE PIF GAINS

The optimal regulator gains and feedforward gains are combined in PIFG to produce the final implementable PIF controller. If METERS is TRUE, PIFG transforms the implementable gains to use units of meters for distance and radians for angular units.

ICH(5,3) (DPIFS) SIMULATE PIF CONTROLLER FOR COMMAND INPUTS

ICH(5,4) (DPIFS) PRINT OUT PLANT STATES AND CONTROLS OF SIMULATION

ICH(5,5) (DPIFS) PRINT OUT STAR TRAJECTORY OF THE SIMULATION

Using information specified by the designer in UMAG, USTRT and USTP (discussed in HDCON), DPIFS simulates the PIF controller in the form used for implementation. A nonlinear representation of the command model is used to generate the command model trajectories. A linear model of the aircraft is used to propagate the plant dynamics. The plant states and controls are printed out if ICH(5,4) is TRUE. The star trajectory computed using the feedforward matrices is printed out if ICH(5,5) is TRUE.

B. DEFINITION OF VARIABLES

All variables used in the PIFCGT main program are defined in Table 1. The common block for each variable is also identified in Table 1.

TABLE 1. PIFCGT Variable Definitions

MATRICES

ACL	WORK MATRIX
AM	STATE,CONTROL CROSS WEIGHTING CAUSED BY QOVE IN CONTINUOUS PLANT
AMHAT	((N+M+L) X M) PIF DISCRETE CROSS WEIGHTING MATRIX
DAC	(3 X M) ACCELEROMETER OUTPUT CONTROL OBSERVATION MATRIX
DX	(N X M) SENSOR OUTPUT CONTROL OBSERVATION MATRIX USE Y=HX*X + DX*U FOR FEEDBACK
ER,EI,IE	IF ICH(2,16) IS FALSE THEN PIFEIG ASSIGNS CLOSED LOOP EIGENVALUES TO THOSE DICTATED BY ER AND EI ER IS THE REAL PART, EI IS THE IMAGINARY PART IE(I) = 1 REPLACE THE ITH EIGENVALUE OF F WITH THOSE OF ER(I),EI(I). THIS FEATURE IS NOT USED. IE(I) = 0 DO NOT REPLACE THE EIGENVALUE OF F
ERV,IEV	IF ICH(2,16) IS FALSE THEN PIFEIG ASSIGNS CLOSED LOOP EIGENVALUES TO THOSE DICTATED BY ERV. EIGENVECTORS ARE STORED AS REAL NUMBERS IN ERV IN ROWS. IF AN EIGENVECTOR IS COMPLEX THE FIRST ROW FOR THE EIGENVECTOR IS THE REAL PART AND THE SECOND ROW IS THE POSITIVE IMAGINARY PART. IEV(I) = 1 REPLACE THE ITH EIGENVECTOR OF F WITH THOSE OF ERV(I). THIS FEATURE IS NOT USED. IEV(I) = 0 DO NOT REPLACE THE EIGENVECTOR OF F
GAMA	((N+M+L) X M) PIF PLANT CONTROL MATRIX
GK	(M X (N+M+L)) PIF GAIN FROM RICCATI EQUATION SOLUTION
HAC	(3 X N) STATE OBSERVATION PERTURBATION MATRICES FOR AX,AY,AZ
HX	(N X N) ACCELEROMETER OUTPUT STATE OBSERVATION MATRIX
PHI	((N+M+L) X (N+M+L)) PIF PLANT SYSTEM MATRIX
PHICL	((N+M+L) X (N+M+L)) CLOSED-LOOP PIF PLANT IN DREG AND IDEAL MODEL MATRIX IN PIFEIG,PIFPLC
QHAT	((N+M+L) X (N+M+L)) PIF DISCRETE STATE QUADRATIC WEIGHT MATRIX
RHAT	(M X M) PIF DISCRETE CONTROL QUADRATIC
W9	WORK MATRIX
X	((N+M+L) X (N+M+L)) PIF DISCRETE RICCATI EQUATION SOLUTION

MATRIX SYSTEMS

FM,GM,HM,DM	QUAD SYSTEM MATRICES FOR COMMAND MODEL (CONTINUOUS MODEL)
DFM,DGM,HM,DM	QUAD SYSTEM MATRICES FOR COMMAND MODEL (DISCRETE MODEL)
F,G,H,D	QUAD SYSTEM MATRICES FOR AIRCRAFT MODEL (CONTINUOUS MODEL)
DF,DG,H,D	QUAD SYSTEM MATRICES FOR AIRCRAFT MODEL (DISCRETE MODEL)
S11,S12,S21,S22	CGT FEEDFORWARD MATRIX SOLUTION (SEE CGTPIF)
S31,S32,S41,S42	PIF REFORMATED GAINS (SEE PIFG AND DPIFS)

TABLE 1. (CONTINUED) PIFCGT Variable Definitions

AERODYNAMIC DATA

CLBDOT	CLADOT	CMBDOT	CMADOT	CNBDOT	CNADOT
CXO	CXP	CXQ	CXR	CXB	CXA
CXDT	CXDE	CXDA	CXDR		
CYO	CYP	CYQ	CYR	CYB	CYA
CYDT	CYDE	CYDA	CYDR		
CZO	CZP	CZQ	CZR	CZB	CZA
CZDT	CZDE	CZDA	CZDR		
CLO	CLP	CLQ	CLR	CLB	CLA
CLDT	CLDE	CLDA	CLDR		
CMO	CMP	CMQ	CMR	CMB	CMA
CMDT	CMDE	CMDA	CMDR		
CNO	CNP	CNQ	CNR	CNB	CNA
CNDT	CNDE	CNDA	CNDR		

OTHER VARIABLES

VARIABLE	COMMON BLOCK	DESCRIPTION
A	AERO	WING AREA (SQ FT)
AK1	RNINFO1	FEEDBACK GAIN OF MODEL
AK2	RNINFO1	FEEDBACK GAIN OF MODEL
B	AERO	WING SPAN (FT)
C	AERO	CBAR MEAN AERODYNAMIC CHORD(FT)

**ALL INPUT QUADRATIC WEIGHTS ARE SQUARED IN CFWM WHEN THEY ARE PLACED IN Q, R, RD, QZ, OR QDVE.

DA	QWHT	R WEIGHT ON AILERON POSITION
DARD	QWHT	RD WEIGHT ON AILERON RATE
DAO	AERO	AILERON TRIM VALUE
DEBUMA	DEBG	T - PRINTS OUT NAMELIST AERO1
DEBUM1	DEBG	T - PRINTS OUT NAMELIST NAM1
DEBUM2	DEBG	T - PRINTS OUT NAMELIST NAM2
DEBUM3	DEBG	T - PRINTS OUT NAMELIST NAM3
DEBUM4	DEBG	T - PRINTS OUT NAMELIST NAM4
DEBUM6	DEBG	T - PRINTS OUT NAMELIST NAM6
DEBU14	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN FGAERO
DEBU22	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN HDCON
DEBU24	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN CFWM
DEBU25	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN PIFMODL
DEBU26	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN REMAT
DEBU27	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN PIFEIG
DEBU31	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN DISCMOD
DEBU42	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN QRMP1
DEBU43	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN QRMHAT
DEBU44	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN PIFPLC
DEBU46	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN DREG
DEBU48	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN PIFFRE
DEBU49	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN PIFG
DEBU53	DEBG	T - PRINTS OUT INTERMEDIATE VALUES IN DPIFS
DELT	EIGEN	CONTROL LAW SAMPLING INTERVAL
DER	QWHT	R WEIGHT ON ELEVATOR POSITION
DERD	QWHT	RD WEIGHT ON ELEVATOR RATE
DEO	AERO	ELEVATOR TRIM VALUE
DREAL	RNINFO	STABILITY TEST PARAMETER IN ASYMREG. SHOULD BE LESS THAN ZERO.

TABLE 1. (CONTINUED) PIFCGT Variable Definitions

DRR	QWHT	R WEIGHT ON RUDDER POSITION
DRRD	QWHT	RD WEIGHT ON RUDDER RATE
DRO	AERO	RUDDER TRIM VALUE
DTH	QWHT	R WEIGHT ON THROTTLE POSITION
DTHD	QWHT	RD WEIGHT ON THROTTLE RATE
DTO	AERO	THROTTLE TRIM VALUE
ECGT	RNINFO	CONVERGENCE TEST IN SOLVER IN CGT FOR INVERTING MATRIX RECOMMAND 1.0E-10 FOR CDC MACHINE
ELOG	RNINFO	CONVERGENCE TEST FOR LOGX1. SHOULD BE 1.0E-10
EPS	EIGEN	1.0E-08 NOT USED
GKOGM	RNINFO1	FEEDBACK GAIN USED TO ADJUST GM SO THAT STEADY STATE SIDESLIP IS ZERO IF ICHOSE = 9
GKDICP	RNINFO2	INTERCEPT GAIN FOR COMPUTING DO FEEDBACK GAIN
GKDMAX	RNINFO2	MAXIMUN DO FEEDBACK GAIN VALUE
GKDMIN	RNINFO2	MINIMUM DO FEEDBACK GAIN VALUE
GKDSLP	RNINFO2	SLOPE GAIN FOR COMPUTING DO FEEDBACK GAIN
GKPSI	RNINFO1	PSI ERROR GAIN IN ICHOSE=6,15 AND 16 DESIGN
GKY	RNINFO1	YERROR GAIN IN ICHOSE = 15 AND 16 DESIGN
GR	AERO	GRAVITY (FT/SEC**2)
ICGT	RNINFO	
ICGT(1)=1		O11, O12, O21, O22 CONSTRUCTED FM,GM,HM,DM,S11,S12,S21,S22,FW,GW,A,B,C,E,S31,S32,S41,S42 MUST APPEAR IN CALLING SEQUENCE BUT ARE NOT USED IN PROGRAM EXECUTION
ICGT(1)=2		S11,S21 CONSTRUCTED S21,S22,FW,GW,A,B,C,E,S31,S32,S41,S42 MUST APPEAR IN CALLING SEQUENCE BUT ARE NOT USED IN PROGRAM EXECUTION
ICGT(1)=3		S11,S21,S21,S22 CONSTRUCTED FW,GW.A,B,C,E,S31,S32,S41,S42 MUST APPEAR IN CALLING SEQUENCE BUT ARE NOT USED IN PROGRAM EXECUTION
ICGT(1)=4		S31,S32,S41,S42 CONSTRUCTED FM,GM,HM,DM,S11,S12,S21,S22 MUST APPEAR IN CALLING SEQUENCE BUT ARE NOT USED IN PROGRAM EXECUTION
ICGT(1)=5		CONSTRUCT ALL FEEDFORWARD MATRICES
ICGT(4)=1		PRINT OUT O11,O12,O21,O22 MATRICES
ICGT(4)=2		PRINT OUT O11,O12,O21,O22,S11,S12,S21,S22 MATRICES
ICGT(5)=1		DO NOT PERFORM QUAD INVERSE O11,O12,O21,O22 ARE PASSED IN CALLING SEQUENCE

TABLE 1. (CONTINUED) PIFCGT Variable Definitions

ICH	EIGEN	LOGICAL ARRAY CONTAINING INFORMATION ON WHICH SUBROUTINES TO CALL FROM PIFCGT
ICHOSE	RNINFO	
ICHOSE = 1		BETA AND ROLL CONSTANT COMMAND SYSTEM FOR LATERAL DYNAMICS. BETA AND ROLL ANGLE ARE COMMANDED IN UMAG(1,1) AND UMAG(2,1), RESPECTIVELY.
ICHOSE = 3		ROLL AND PSI TURNING COMMAND SYSTEM FOR LATERAL SYSTEM. PSI IS COMMANDED IN UMAG(1,1).
ICHOSE = 4		PITCH COMMAND SYSTEM FOR LONGITUDINAL DYNAMICS. PITCH ANGLE IS COMMANDED IN UMAG(1,1).
ICHOSE = 6		PSI OUTER LOOP COMMAND SYSTEM USING ROLL AND RUDDER INNER LOOP - GKPSI IS THE OUTER LOOP YAW GA*N. YAW AND RUDDER COMMANDED IN UMAG(1,1) AND UMAG(2,1*.
ICHOSE = 7		Z,ZDOT COMMAND SYSTEM FOR LONGITUDINAL SYSTEM WITH Z COMMANDED IN UMAG(1,1).
ICHOSE = 9		LOCALIZER TRACKING MODEL COMMANDING Y AND ROLL. Y INITIAL CONDITION IS SET IN UMAG(1,1). PSI INTERCEPT ANGLE IS SET IN UMAG(2,1). USTRT MUST START AT ZERO USTP MUST BE AT MAXMIUM.
ICHOSE = 12		ROLL AND RUDDER COMMAND SYSTEM FOR LATERAL SYSTEM. ROLL AND RUDDER COMMANDED IN UMAG(1,1) AND UMAG(2,1).
ICHOSE = 14		D COMMAND SYSTEM FOR GLIDESLOPE TRACKING. UMAG(1,1) CONTAINS D POSITION OFFSET AT START AT OUTER MARKER. USTRT SHOULD BE 0.0. USTOP SHOULD BE MAXIMUM TIME. A MODEL IN SIMULATION MOVES D TO ZERO KEEPING FLIGHT PATH ANGLE WITHIN BOUNDS. Z IN F AND G IS REPLACED WITH D DYNAMICS. LAMDA IS FLIGHT PATH ANGLE IN DEGREES. ZETA IS GROUND RELAIVE HEADING ANGLE IN DEGREES.
ICHOSE = 15		Y OUTER LOOP COMMAND SYSTEM USING ROLL AND RUDDER INNER LOOP - GKPSI IS THE OUTER LOOP YAW GAIN AND GKY IS THE OUTER LOOP Y GAIN. Y INITIAL CONDITION IS SET IN UMAG(1,1). PSI INTERCEPT ANGLE IS SET IN UMAG(2,1). USTRT MUST START AT ZERO USTP MUST BE AT MAXMIUM.
ICHOSE = 16		Y OUTER LOOP COMMAND SYSTEM USING ROLL AND RUDDER INNER LOOP - GKY IS THE OUTER LOOP Y GAIN. Y INITIAL CONDITION IS SET IN UMAG(1,1). PSI INTERCEPT ANGLE IS SET IN UMAG(2,1). USTRT MUST START AT ZERO USTP MUST BE AT MAXMIUM.
IOREG	RNINFO	OPTIONS FOR ASYMREG AS DISCUSSED IN ORACLES
IFREEF	EIGEN	0, 1 OR 2 EIGENVALUE/EIGENVECTORS FOR F MATRIX
IFREEG	EIGEN	0, 1 OR 2 NOT USED
IFREEP	EIGEN	0, 1 OR 2 EIGENVALUES/EIGENVECTORS FOR PIF CLOSED-LOOP
IFREIG	EIGEN	0, 1 OR 2 EIGENVALUES/EIGENVECTORS FOR PIFEIG IDEAL MODEL MATRIX DESIGN
	0 -	NO CALCULATIONS IN FREEQ
	1 -	EIGENVALUES ONLY
	2 -	EIGENVALUES AND EIGENVECTORS
IKDGM	RNINFO1	NUMBER OF ITERATIONS ALLOWED TO ADJUST DGM SO STEADY STATE SIDESLIP IS 0.0 IF ICHOSE =9,15 OR 16
IMAX	LABEL	NUMBER OF TIME STEPS IN OPIFS SIMULATION
IRUN	LABEL	NUMBER OF SIMULATIONS IN OPIFS (SHOULD BE 1.0)
ITER	RNINFO	MAXIMUM NUMBER OF ITERATIVE RESIDUALTIONS IN SOLVER

TABLE 1. (CONTINUED) PIFCGT Variable Definitions

IX	AERO	MOMENT OF INERTIA (SLUG/FT**2)
IY	AERO	MOMENT OF INERTIA (SLUG/FT**2)
IZ	AERO	MOMENT OF INERTIA (SLUG/FT**2)
L	GLAD	NUMBER OF STARTING COMMAND VARIABLES
LAMDA	RNINFO	FLIGHT PATH ANGLE OF TRIMMED FLIGHT CONDITION FOR AERODYNAMIC DATA (USUALLY 0.0 STRAIGHT AND LEVEL FLIGHT)
LAMDAGS	RNINFO	GLIDESLOPE ANGLE (ALWAYS POSITIVE USUALLY 3.0 DEG)
LAMDA1	RNINFO	FLIGHT PATH ANGLE OF TRIMMED FLIGHT CONDITION FOR AERODYNAMIC DATA (USUALLY 0.0 STRAIGHT AND LEVEL FLIGHT)
M	GLAD	NUMBER OF STARTING CONTROLS VARIABLES
METERS	RNINFO	LOGICAL VARIABLE THAT DETERMINES IF THE SIMULATION DATA IS IN METERS OR NOT METERS = TRUE (DATA IN METERS/SEC) METERS = FALSE (DATA IN FEET/SEC) NEW CONTROL GAINS ARE PRINTED OUT IN PIFG IF METERS IS TRUE. THE NEW CONTROL GAINS ASSUME RADIANS FOR ANGLES AND CONTROL VARIABLES AND METERS FOR DISTANCE VARIABLES.
MLOG	RNINFO	MAXIMUM NUMBER OF ITERATIONS IN LOG OF A MATRIX (LOGX1) SUBROUTINE
MM	GLAD	DIMENSION OF MODEL CONTROLS
N	GLAD	STARTING NUMBER OF STATE VARIABLES
NM	GLAD	DIMENSION OF MODEL
PBDQ	QWHT	QDVE WEIGHTING OF PDOT
PBQ	QWHT	Q WEIGHTING OF P
PHDQ	QWHT	QDVE WEIGHTING OF PHIDOT
PHIO	AERO	TRIM VALUE OF PHI
PHQ	QWHT	Q WEIGHTING OF PHI
PSDQ	QWHT	QDVE WEIGHTING OF PSIDOT
PSIO	AERO	TRIM VALUE OF PSI
PSQ	QWHT	Q WEIGHTING OF PSI
PO	AERO	TRIM VALUE OF P
QBDQ	QWHT	Q WEIGHTING OF QDOT
QBQ	QWHT	Q WEIGHTING OF Q
QO	AERO	TRIM VALUE OF Q
RBDQ	QWHT	QDVE WEIGHTING OF RDOT
RBQ	QWHT	Q WEIGHTING OF R
RHO	AERO	AIR DENSITY (SLUGS/FT**3)
RO	AERO	TRIM VALUE OF R
S	AERO	CALCULATED IN FGAERO
SS	EIGEN	SS = 1.0 IF ICGT(1) .GT. 2 AND PLANT, MODEL, AND DISTURBANCE ARE DISCRETE TIME MODELS SS = 0.0 IF ICGT(1) .GT. 2 AND PLANT, MODEL, AND DISTURBANCE ARE CONTINUOUS TIME MODELS SS = ANY NUMBER IF ICGT(1) .LE. 2 EXCEPT PLANT TRANSMISSION ZEROES
THDQ	QWHT	Q WEIGHTING OF THETADOT
THETAO	AERO	TRIM VALUE OF THETA
THQ	QWHT	Q WEIGHTING OF THETA
TOL	RNINFO	CONVERGENCE CRITERIA FOR ITERATIVE RESIDUALTION IN SOLVER. RECOMMEND 1.0E-10 FOR CDC MACHINE
UBDQ	QWHT	QDVE WEIGHTING OF UDOT
UBQ	QWHT	Q WEIGHTING OF U
UL	LABEL	ARRAY CONTAINING THE LABLES FOR THE CONTROLS
UMAG	UDIR	ARRAY OF MAGNITUDE OF COMMAND INPUTS IN DPIFS SIMULATION (MM X IRUN) NOT REDUCED IN REMAT
UMR	EIGEN	LOGICAL ARRAY WITH INFORMATION TO REFORMAT THE COMMAND MODEL CONTROL MATRIX GM (L X MM)
UR	EIGEN	LOGICAL ARRAY CONTAINING THE INFORMATION TO REFORMAT THE AIRCRAFT CONTROL MATRIX-G (NXM)
USTP	UDIR	ARRAY OF TIMES AT WHICH COMMAND INPUTS ARE RETURNED TO ZERO IN DPIFS SIMULATION (MM X IRUN) NOT REDUCED IN REMAT

TABLE 1. (CONCLUDED) PIFCGT Variable Definitions

USTRT	UDIR	ARRAY OF TIMES AT WHICH COMMAND INPUTS ARE CHANGED FROM ZERO TO UMAG VALUES IN OPIFS SIMULATION (MM X IRUN) NOT REDUCED IN REMAT
UXMAG	LABEL	ARRAY CONTAINING THE SCALING VALUES FOR CONTROLS (NOT USED)
UO	AERO	TRIM VALUE OF U
VBDQ	QWHT	QDVE WEIGHTING OF VDOT
VHQ	QWHT	Q WEIGHTING OF V
VTO	AERO	TRIM VALUE OF VT
VTP	RNINFO	TOTAL AIRSPEED VELOCITY IN FT/SEC**2. USED IN INTERP. TRIM VALUES AND AERODYNAMIC DATA ARE INTERPOLATED SO THAT AIRCRAFT MODEL IS FOUND FOR VTO = VTP.
VO	AERO	TRIM VALUE OF V
WBDQ	QWHT	QDVE WEIGHTING OF WDOT
WBQ	QWHT	Q WEIGHTING OF W
WT	AERO	WEIGHT (LBS)
WO	AERO	TRIM VALUE OF W
XIDQ	QWHT	WEIGHTING ON XDOT POSITION
XIQ	QWHT	WEIGHTING ON X POSITION
XL	LABEL	ARRAY CONTAINING THE LABELS FOR THE STATES
XMR	EIGEN	LOGICAL ARRAY WITH INFORMATION TO REFORMAT THE COMMAND MODEL SYSTEM MATRIX FM (NM X NM)
XR	EIGEN	LOGICAL ARRAY WITH INFORMATION TO REFORMAT THE AIRCRAFT SYSTEM MATRIX F (N X N)
XS	HDXA	LOCATION OF ACCELEROMETERS IN BODY AXIS, X-POSITION WITH REFERENCE TO CG
XXMAG	LABEL	ARRAY CONTAINING THE SCALING VALUES FOR STATES USED IN OPIFS WHEN METERS IS TRUE, AUTOMATICALLY SCALED
YIDQ	QWHT	WEIGHTING ON YDOT POSITION
YIQ	QWHT	WEIGHTING ON Y POSITITON
YL	LABEL	ARRAY CONTAINING THE LABELS FOR THE COMMANDS
YR	EIGEN	LOGICAL ARRAY WITH INFORMATION TO REFORMAT THE OUTPUT MATRICES H,D,HM,DM (L X N),(L X M),(L X NM) (L X MM)
YS	HDXA	LOCATION OF ACCELEROMETERS IN BODYAXIS, Y-POSITION WITH REFERENCE TO CG
YXMAG	LABEL	ARRAY CONTAINING THE SCALING VALUES FOR OUTPUTS (NOT USED)
ZCDMAX	RNINFO1	MULTIPLICATION FACTOR FOR CHANGING ZDD MAXIMUM
ZCLMAX	RNINFO1	MULTIPLICATION FACTORS FOR CHANGING ZCLOSE
ZCLMIN	RNINFO1	MULTIPLICATION FACTORS FOR CHANGING ZCLOSE
ZCLOSE	RNINFO1	POSITION DIFFERENCE WHEN FULL DECELERATION BEGINS
ZDDMAX	RNINFO1	MAXIMUM ACCELEARTION
ZDOMAX	RNINFO1	MAXIUM MODEL RATE
ZETA	RNINFO	VELOCITY VECTOR HEADING ANGLE OF TRIMMED FLIGHT CONDITION (USUALLY 0.0)
ZETA1	RNINFO	VELOCITY VECTOR HEADING ANGLE OF TRIMMED FLIGHT CONDITION (USUALLY 0.0)
ZIDQ	QWHT	WEIGHTING ON ZDOT POSITION
ZIQ	QWHT	WEIGHTING ON Z POSITION
ZS	HDXA	LOCATION OF ACCELEROMETERS IN BODYAXIS, Z-POSITION WITH REFERENCE TO CG
Z1Q	QWHT	QZ WEIGHTING OF FIRST INTEGRATOR STATE (QZ REFORMATED USING YR)
Z2Q	QWHT	QZ WEIGHTING OF SECOND INTEGRATOR STATE (QZ REFORMATED USING YR)
Z3Q	QWHT	QZ WEIGHTING OF THIRD INTEGRATOR STATE (QZ REFORMATED USING YR)
Z4Q	QWHT	QZ WEIGHTING OF FOURTH INTEGRATOR STATE (QZ REFORMATED USING YR)

III. SUBROUTINE DOCUMENTATION

All subroutines used by PIFCGT are from ORACLS (ORACLIB in Appendix B) or from the PIF library (PIFLIBO in Appendix B). This chapter individually documents in alphabetical order the PIF library subroutines. The documentation follows the same format at ORACLS in Ref. 2.

A. BLOCK DATA (AEROD)

1. PURPOSE

Block data program AEROD initializes aerodynamic data in common blocks AERO and AER1. Other variables are initialized from common blocks listed in 2d below.

2. USAGE

Under the NOS 1.3 operating system, AEROD is placed in the submit file as shown in Appendix B.

a. Calling Sequence

None

b. Input Arguments

None

c. Output Arguments

None

d. Common Blocks

RNINFO, LABEL, EIGEN, GLAD, IDENTS, DEBUG, AERO, AER1, HDXA

e. Error Messages

None

f. Subroutines Employed by AEROD

None

g. Subroutines Employing AEROD

None

h. Concluding Remarks

None

B. SAMPLED-DATA REGULATOR (ALPHA AND BETA)

1. PURPOSE

The purpose of the subroutines ALPHA and BETA is to compute the following functions depending on the values of LLL and H,

If LLL = 2, and H = 0.0 then compute

$$\text{SUM} = \int_0^{\Delta t} \int_0^{\tau} e^{A^T s}\, ds\; Q \int^{\tau} e^{As}\, ds\, d\tau \tag{1}$$

If LLL = 1, and H = -1.0 then compute

$$\text{SUM} = \int_0^{\Delta t} e^{A^T \tau}\; Q \int_0^{\tau} e^{As}\, ds\, d\tau \tag{2}$$

Subroutine ALPHA calls subroutine BETA as part of its execution.

2. USAGE

a. Calling Sequence

CALL ALPHA (N, A, AT, ASIG, SIG, O, GEST, EPSLO, LLL, H, SUM, A1, SS1,
SS, A2, BVEC, AVEC, XMLIH)

b. Input Arguments

N — Integer scalar value indicating matrix dimension; A, AT, ASIG, Q, SUM, A1, SS1, SS, A2, BVEC, AVEC, XMLIH - all N x N.

A — Matrix packed by column in one-dimensional array; not destroyed upon return.

AT — Matrix packed by column in one-dimensional array; not destroyed upon return. AT must contain the transpose of A.

ASIG, A1, SS1, SS, A2, BVEC, AVEC, XMLIH — Working space vectors of dimension at least N x N

SIG — Real scalar indicating the sampling time Δt.

GEST — An index for starting the test of convergence in computing the integrals. A value of 5.0 is recommended.

EPSLO Real scalar indicating the convergence criterion in computing the integrals. A value of 0.00001 is recommended.

LLL, H An integer scalar and a real scalar, respectively, used to determine the integrals discussed in the PURPOSE section.

c. Output Arguments

SUM An N x N matrix packed by columns into a one-dimensional array containing the evaluation of the computed integrals.

d. Common Blocks

None

e. Error Messages

None

f. Subroutines Employed by ALPHA and BETA

ORACLS - MULT, UNITY

g. Subroutines Employing ALPHA and BETA

PIFLIB - QRMHAT

h. Concluding Remarks

None

C. SOLUTION OF A*Y*B-X=C (AXBMXC)

1. PURPOSE

Subroutine AXBMXC solves the algebraic matrix equation

$$A*X*B - X = C \tag{1}$$

First the matrix A is transformed to the lower real SCHUR form using the unitary transformation U. Next, the matrix B is transformed to upper real SCHUR form using the unitary transformation V. The matrix C is transformed using $U^T * C * V$. Equation 1 reduces to

$$(U^TAU)*(U^TXV)*(V^TBV)-(U^TXV) = U^TCV \tag{2}$$

or

$$A_s * X_s * B_s - X_s = C_s \tag{3}$$

The subroutine SHRSOL is used to solve Eq. 3 for X_s. The solution X, is reconstructed for X_s as follows

$$X = U*X_s*V^T \tag{4}$$

It is assumed that

$$\lambda_i^A \lambda_j^B \neq 1 \quad (i=1,2,\ldots,N; \quad j=1,2,\ldots,M) \tag{5}$$

where λ_i^A and λ_j^B are eigenvalues of A and B respectively. If Eq. 5 is true, Eq. 1 has a unique solution for X.

2. USAGE

a. Calling Sequence

CALL AXBMXC (A,U,B,V,C,N,NA,NU,M,NB,NV,NC,EPSA,EPSB,FAIL)

b. Input Arguments

A,B,C   Matrices stored in two-dimensional array. First dimensions are NA, NB, and NC, respectively. Second dimensions are at least N, M, and N, respectively. Destroyed upon return.

N — Order of the matrix A

M — Order of the matrix B

NA,NB,NC,
NU,NV — Maximum first dimensions of the matrices A, B, C, U, and V, respectively, as given in the dimension statement of the calling program. It is required that NA .GT. N+1, NB .GT. M+1, NC .GT. N+1, NU .GT. N+1, and NV .GT. M+1

EPSA — Criterion for zeroing elements in the matrix A in the subroutine SCHUR. A(I,J) is considered effectively 0 if it is less than EPSA times the infinity norm of the upper Hessenberg form of A

NOTE: If EPSA .LT. 0 it is assumed that A is already in lower real SCHUR form and U is the appropriate transformation matrix.

EPSB — Criterion for zeroing elements in the matrix B in the subroutine SCHUR. B(I,J) is considered effectively 0 if it is less than EPSA times the infinity norm of the upper Hessenberg form of B.

NOTE: If EPSB .LT. 0 it is assumed that B is already in upper real SCHUR form and V is the appropriate transformation matrix.

c. Output Arguments

FAIL — Integer flag indicating success of reduction of A and B to lower and upper real SCHUR form. If FAIL is not zero, SCHUR was unsuccessful.

U — Two-dimensional array containing the orthogonal matrix which reduces A to lower real SCHUR form. First dimension is NU.

Second dimension is at least N.

V Two-dimensional array containing the orthogonal matrix which reduces B to upper real SCHUR form. First dimension is NV. Second dimension is at least M.

A,B Two-dimensional arrays containing the original A and B matrices transformed to lower and upper real SCHUR form, respectively.

C Two-dimensional array containing the solution, X.

d. COMMON Blocks

None

e. Error Messages

None

f. Subroutines Employed by AXBMXC

ORACLS - BCKMLT, HSHLDR, SCHUR, SYSSLV

PIFLIB - SHRSOL

g. Subroutines Employing AXBMXC

PIFLIB - SOLVER

h. Concluding Remarks

It is recommended that the program SOLVER, which calls AXBMXC, be used to solve Eq. 1. This subroutine is a generalization of the algorithm reported in Ref. 3.

D. CREATE WEIGHTING MATRICES (CFWM)

1. PURPOSE

The subroutine CFWM constructs the diagonal weighting matrices used in the PIF control law quadratic cost function. The state weights UBQ, WBQ, QBQ, THQ, XIQ, ZIQ, VHQ, RBQ, PBQ, PHQ, PSQ, YIQ are squared and placed along the Q matrix diagonal. The control weights DTH, DER, DA, and DRR are squared and placed along the R matrix diagonal. The control rate weights DTHD, DERD, DARD, DRRD are squared and placed along the RD matrix diagonal. The integrator weights Z1Q, Z2Q, Z3Q and Z4Q are squared and placed along the QZ matrix diagonal. The state rate weights UBDQ, WBDQ, QBDQ, THDQ, XIDQ, ZIDQ, VBDQ, RBDQ, PBDQ, PHDQ, PSDQ and YIDQ are squared and placed along the W9(LW1) diagonal. The following matrix is constructed

$$\left[\begin{array}{c|c} Q1 & AM^T \\ \hline AM & R1 \end{array}\right] = \begin{bmatrix} F^T \\ G^T \end{bmatrix} W9(LW1) \begin{bmatrix} F & G \end{bmatrix} \tag{1}$$

The Q1 matrix is added to Q and the R1 matrix is added to R completing the weighting matrix constructions. F and G are the continuous-time plant system matrix and control effect matrix, respectively.

2. USAGE

a. Calling Sequence

CALL CFWM(F, NF, G, NG, Q, NQ, R, NR, AM, NAM, RD, NRD, QZ, NQZ, W9, NW9DIM, DEBUG)

b. Input Arguments

F, G Matrices packed by columns into one-dimensional arrays; not destroyed upon return.

NF, NG, NQ,

NR, NAM, NRD,

NQZ Two dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N. Integer values must be: NF(12, 12), NG(12, 4), NQ(12, 12), NR(4, 4), NAM(4, 12,), NRD(4, 4), NQZ(12, 12).

W9 Work matrix packed by columns into a one-dimensional array. Dimension must be at least 4 * ((NF(1) + NG(2))**2 + 1).

DEBUG Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug information

NW9DIM Integer scalar containing the maximum dimension of the W9 matrix.

c. Output Arguments

Q, R, AM,

RD, QZ Weighting matrices packed by columns into one dimensional arrays.

d. COMMON Blocks

QWHT

e. Error Messages

If the W9 matrix is insufficiently large the message "THE W9 MATRIX IS SMALL" is printed and the program stops. If DEBUG is true then Q, R, RD, QZ, and W9(LW1) (QDVE) are printed after being diagonally constructed. After the state rate construction and assimilation Q, R, RD, QZ, and W9(LW1) (QDVE) are printed again along with AM.

f. Subroutines Employed by CFWM

ORACLS - JUXTC, ADD, PRNTITL, DIAGPUT, PRNT, MATBA, EXTR

g. Subroutines Employing QRMHAT

PIFLIB - PIFCGT

h. Concluding Remarks

None

E. FEEDFORWARD MATRICES (CGTPIF)

1. PURPOSE

Subroutine CGTPIF computes the feedforward matrices S_{11}, S_{12}, S_{21}, S_{22}, S_{31}, S_{32}, S_{41}, and S_{42}. The subroutine determines the feedforward matrices by solving the following matrix equation

$$\begin{bmatrix} (F-ssI) & G \\ H & D \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} & S_{31} & S_{32} \\ S_{21} & S_{22} & S_{41} & S_{42} \end{bmatrix} = \begin{bmatrix} S_{11}(F_m-ssI) & S_{11}G_m & S_{31}(F_w-ssI)-AS_{31}G_w-B \\ H_m & D_m & -C & -E \end{bmatrix} \quad (1)$$

The feedforward matrix equation is valid for both continuous-time models and discrete-time models. The scalar ss in Eq. 1 is chosen by the designer under the following restrictions. ss cannot equal a transmission zero of the plant. If the dynamical models are representations of continuous-time systems and S_{12}, S_{22}, S_{32}, or S_{42} is to be determined, then ss must be zero. If the dynamical models are representations of discrete-time systems and S_{12}, S_{22}, S_{32} or S_{42} is to be determined then ss must be one.

The plant, disturbance, and model matrices in the feedforward matrix equations are assumed to satisfy the following dynamical equations

CONTINUOUS TIME

PLANT

$$\dot{\underline{x}} = F\underline{x} + G\underline{u} + A\underline{w} + B\underline{s} \quad (2)$$

$$\underline{y} = H\underline{x} + D\underline{u} + C\underline{w} + E\underline{s} \quad (3)$$

DISTURBANCE TO BE REJECTED

$$\dot{\underline{w}} = F_w\underline{w} + G_w\underline{s} \quad (4)$$

MODEL TO BE FOLLOWED

$$\dot{\underline{x}}_m = F_m\underline{x}_m + G_m\underline{u}_m \tag{5}$$

$$\underline{y}_m = H_m\underline{x}_m + D_m\underline{u}_m \tag{6}$$

DISCRETE TIME

PLANT

$$\underline{x}_{k+1} = F\underline{x}_k + G\underline{u}_k + A\underline{w}_k + B\underline{s}_k \tag{7}$$

$$\underline{y}_k = H\underline{x}_k + D\underline{u}_k + C\underline{w}_k + E\underline{s}_k \tag{8}$$

DISTURBANCE TO BE REJECTED

$$\underline{w}_{k+1} = F_w\underline{w}_k + G_w\underline{s}_k \tag{9}$$

MODEL TO BE FOLLOWED

$$\underline{x}_{m,k+1} = F_m\underline{x}_{m,k} + G_m\underline{u}_{m,k} \tag{10}$$

$$\underline{y}_{m,k} = H_m\underline{x}_{m,k} + D_m\underline{u}_{m,k} \tag{11}$$

The program begins by forming the quad partition matrix on the left side of Eq. 1. If the number of outputs in $\underline{y}$ equal the number of controls in $\underline{u}$, the program attempts to invert the quad partition matrix using GELIM. If the quad partition matrix is not square or GELIM encountered an error in attempting to invert the quad partition matrix, CGTPIF attempts to find the pseudo-inverse of the quad partition matrix using singular value decomposition (SNVDEC). If SNVDEC also encounters an error, a message is printed and the program continues.

After obtaining the inverse (or pseudo-inverse)

$$\begin{bmatrix} (F-ssI) & G \\ H & D \end{bmatrix}^{-1} = \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} \tag{12}$$

Eq. 1 is separated into eight algebraic matrix equations,

$$S_{11} = \Omega_{11} \; S_{11}(F_m - ssI) + \Omega_{12} \; H_m \tag{13}$$

$$S_{12} = \Omega_{11} \; S_{11} \; G_m + \Omega_{12} \; D_m \tag{14}$$

$$S_{21} = \Omega_{21} \; S_{11}(F_m - ssI) + \Omega_{22} \; H_m \tag{15}$$

$$S_{22} = \Omega_{21} \; S_{11} \; G_m + \Omega_{22} \; D_m \tag{16}$$

$$S_{31} = \Omega_{11} \; S_{31}(F_w - ssI) + \Omega_{11}A - \Omega_{12}C \tag{17}$$

$$S_{32} = \Omega_{11} \; S_{31} \; G_w - \Omega_{11}B - \Omega_{12}E \tag{18}$$

$$S_{41} = \Omega_{21} \; S_{31}(F_w - ssI) + \Omega_{21}A - \Omega_{22}C \tag{19}$$

$$S_{42} = \Omega_{21} \; S_{31} \; G_m - \Omega_{21}B - \Omega_{22}E \tag{20}$$

Equations 13 and 17 are solved using SOLVER. Equations 14, 15, 16, 18, 19, and 20 are solved using matrix multiplication.

2. USAGE

a. Calling Sequence

CALL CGTPIF (F,G,H,D,A,B,C,E,FW,GW,FM,GM,HM,DM,W1,W2,W3,W4,W5,O11,O12, O21,O22,S11,S12,S21,S22,S31,S32,S41,S42,N,M,L,NM,MM,NW,MW,TOL,ITER,SS, ICGT,ECGT,NS11,NS12,NS21,NS22,NS31,NS32,NS41,NS42)

b. Input Arguments

F,G,H,D, A,B,C,E, FW,GW,FM, GM,HM,DM	Matrices packed by columns in one-dimensional arrays; not destroyed upon return.
N,M,L, NM,MM, NW,MW	Integer scalar values indicating matrix dimensions: F-NxN, G-NxM, H-LxN, D-LxM, A-NxNW, B-NxMW, C-LxNW, E-LxMW, FW-NWxNW, GW-NWxMW, FM-NMxNM, GM-NMxMM, HM-LxNM, DM-LxMM
TOL	Real scalar indicating convergence criteria for iterative refinement in SOLVER. Recommend 1.0E-10 for CDC machine.
ITER	Integer scalar indicating maximum number of iterative refinements in SOLVER. Recommend 10.
ECGT	Real scalar indicating convergence criteria in SOLVER for inverting a matrix. Recommend 1.0E-10 for CDC machine.
ICGT	Five dimensional integer option vector:

ICGT(1) = 1	O11, O12, O21, O22 constructed. FM,GM,HM,DM,S11,S12,S22,FW,GW,A,B,C,E, S31,S32,S41,S42 must appear in calling sequence but are not used in program execution.
ICGT(1) = 2	S11, S21 constructed. S21,S22,FW,GW,A,B,C,E,S31,S32,S41,S42 must appear in calling sequence but are not used in program execution.

ICGT(1) = 3 S11, S12, S21, S22 constructed.

FW,GW,A,B,C,E,S31,S32,S41,S42 must appear in calling sequence but are not used in program execution.

ICGT(1) = 4 S31, S32, S41, S42 constructed.

FM,GM,HM,DM,S11,S12,S21,S22 must appear in calling sequence but are not used in program execution.

ICGT(1) = 5 Construct all feedforward matrices.

ICGT(4) = 1 Print out O11, O12, O21, O22 matrices.

ICGT(4) = 2 Print out all matrices.

ICGT(5) = 1 Do not perform quad inverse. O11, O12, O21, O22, are passed in calling sequence.

SS Scalar option

SS = 1.0 if ICGT(1) .GT. 2 and plant, model and disturbance are discrete time models

SS = 0.0 if ICGT(1) .GT. 2 and plant, model and disturbance are continuous time models

SS = Any number if ICGT(1) .LE. 2 except plant transmission zeroes.

W1,W2,W3
W4,W5 Working space vectors of dimension at least:

W1 - (2*(N+MAX(L,M)+2))**2, W2,W3,W4,W5 - (N+MAX(L,M)+2)**2

c. Output Arguments

O11,O12,
O21,O22 Matrices packed by columns into one-dimensional arrays.

Upon normal return and depending on ICGT, the matrices contain the partitioned quad inverse

$$\begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix} = \begin{bmatrix} O11 & O12 \\ O21 & O22 \end{bmatrix}$$

Dimensions of the matrices are at least:

O11 - N**2, O12 - N*L, O21 - M*N, O22 - M*L.

S11,S12,
S21,S22,
S31,S32,
S41,S42 Matrices packed by columns into one-dimensional arrays.

Upon normal return and depending on ICGT, the matrices contain the feedforward matrices. Dimensions of the matrices are at least (N+MAX(NM,NW)+2)**2.

NS11,NS12,
NS21,NS22,
NS31,NS32,
NS41,NS42 Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N. Example,

NS11(1) = N

NS11(2) = NM

d. COMMON Blocks

None

e. Error Messages

If ICGT(1) .GT. 2 and SS .NE. 1 the message "???????????? SS IS INCORRECT IN CGT ???????????? BUT PROGRAM CONTINUES SS = __________." is printed. If the plant is a continuous-time model and SS = 0.0, then the error should be ignored. CGTPIF cannot determine if the dynamic models are continuous or discrete and operates under the implicit assumption that the models are discrete.

If GELIM or SNVDEC encounter an error the message "SNVDEC ERROR IN CGT IERR = __________" is printed.

f. Subroutines Employed by CGTPIF

ORACLS - MULT, NULL, SCALE, SUBT, UNITY, ADD, NORMS, BCKMLT, HSHLDR, SCHUR, SYSSLV, GELIM, DETFAC

PIFLIB - AXBMXC, SHRSOL, PRNT, PRNTITL.

g. Subroutines Employing CGTPIF

PIFLIB - PIFCGT, PIFPLC

h. Concluding Remarks

The theory and use of feedforward matrices are discussed in Refs. 4 and 5.

F. CLOSED-LOOP d MODEL (CLDMOD)

1. PURPOSE

Subroutine CLDMOD propagates the dynamic command model for the APR GS autopilot mode. The subroutine is similiar to the APR GS command model in the flight computer code. A discussion of the equations used in CLDMOD is given in Ref. 1.

2. USAGE

a. Calling Sequence

CALL CLDMOD (D1, DO, LAMDMAX, LAMIN, DEBUG, LAMDAGS, VTP, LAMC, GKDSLP, GKDICP, GKDMAX, GKDMIN)

b. Input Arguments

D1, DO — k and k-1 commanded d perpendicular position of aircraft from glideslope. Changed upon return.

LAMDMAX — The maximum allowed flight path angle change in DELT seconds (ZDDMAX in input NAMELIST)

DELT — The control law sampling interval

LAMAX — The maximum allowed flight path angle. (ZCLMAX in input NAMELIST)

LAMIN — The minimum allowed flight path angle. (ZCLMIN in input NAMELIST)

DEBUG — Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug information

LAMDAGS — The glideslope angle (positive)

VTP — The total airspeed in ft/sec

LAMC The commanded model dynamics flight path angle at k-1. Changed upon return.

GKDSLP The slope gain for computing the DO feedback gain.

GKDICP The intercept gain for computing the DO feedback gain.

GKDMAX The maximum DO feedback gain value.

GKDMIN The minimum DO feedback gain value.

c. Output Arguments

D1, DO k+1 and k commanded d perpendicular position of aircraft from glideslope.

LAMC The commanded model dynamics flight path angle at k.

d. COMMON Blocks

None

e. Error Messages

The first pass through the subroutine causes GK1, LAMAX, LAMIN and LAMDAGS to be printed out. If DEBUG is TRUE, D1, D0, LAMDAGS, GK1, and LAMC are printed out during the course of the simulation.

f. Subroutines Employed by CLDMOD

None

g. Subroutines Employing CLDMOD

PIFLIB - DPIFS

h. Concluding Remarks

None

G. CLOSED-LOOP Y MODEL (CLYMOD)

1. PURPOSE

Subroutine CLYMOD propagates the dynamic command model for the APR LOCI, APR LOCR, and APR LOCP autopilot modes. The subroutine is similiar to the APR LOC command model propagation model in the flight computer code. A discussion of the equations used in CLYMOD is given in Ref. 1.

2. USAGE

a. Calling Sequence

CALL CLYMOD (Y1, Y0, PS11, PS10, YC, PSIC, AK2, ROLLCM, GKDSLP, GKDICP, GKDMAX, GKDMIN, DELT, DEBUG RDCMAX, VTP, ROLLC)

b. Input Arguments

Y1, Y0	k and k-1 commanded Y position of aircraft from localizer beam center line. Changed upon return.
PS11, PS10	k and k-1 commanded yaw position of the aircraft relative to the runway. Changed upon return.
YC	The commanded value of y which is set to 0.0.
PSIC	The maximum allowed intercept angle, usually 45 deg.
AK2	The yaw error feedback gain for the roll angle command.
ROLLCM	The maximum allowed value of the roll angle command during capture (ZDDMAX in input NAMELIST)
GKDSLP	The slope gain for computing the Y0 feedback gain.
GKDICP	The intercept gain for computing the Y0 feedback gain.
GKDMAX	The maximum Y0 feedback gain value.

GKDMIN The minimum YO feedback gain value.

DELT The control law sampling interval.

DEBUG Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug information

RDCMAX The maximum allowed roll angle change in DELT sec (ZCDMAX in input NAMELIST)

VTP The total airspeed in ft/sec

ROLLC The commanded value of roll angle at k-1. Changed upon return.

c. Output Arguments

Y1, YO k+1 and k commanded y position of aircraft from beam center line.

PSI1, PSIO k+1 and k commanded yaw position of aircraft.

ROLLC The commanded value of roll angle at k.

d. COMMON Blocks

None

e. Error Messages

The first pass through the subroutine causes PSIC, AK2, ROLLCM and RDCMAX to be printed out. If DEBUG is TRUE, PSICM, YO, GK2, PSIO, ROLLC, X1, GK1 and PSID are printed out during the course of the simulation.

f. Subroutines Employed by CLYMOD

None

g. Subroutines Employing CLYMOD

PIFLIB - DPIFS

h. Concluding Remarks

None

H. CLOSED-LOOP ψ AND Z MODEL (CLXMOD)

1. PURPOSE

Subroutine CLXMOD propagates the dynamic command model for the HDG SEL and ALT SEL autopilot modes. The subroutine is similiar to the HDG SEL and ALT SEL command model propagation models in the flight computer code. A discussion of the equations used in CLXMOD is given in Ref. 1.

2. USAGE

a. Calling Sequence

CALL CLXMOD (Z1, Z0, ZD1, ZD0, ZC1, ZCD, AK1, AK2, ZDDMAX, ACLOSE ZDOMAX, DELT, ZCLMAX, ZCLMIN, ZCDEAD, DEBUG, ZCDMAX, ICHOSE, VTP0

b. Input Arguments

Z1, Z0	k and k-1 commanded z (Ψ) position of aircraft
ZD1, ZDϕ	k and k-1 commanded $\dot{z}$ (ϕ) of aircraft. .
ZC1	Commanded value of Z(Ψ) position the model is required to track.
ZCD	Commanded value of $\dot{z}$ (ϕ), usually 0.0.
AK1, AK2	Position and velocity feedback gains for the model
ZDDMAX	The maximum allowed model acceleration ($\ddot{z}$ or $\dot{\phi}$)
DELT	The sampling interval of the control law
DEBUG	Logical scalar indicating the following: TRUE: Print out debug information FALSE: Do not print out debug information
ICHOSE	Integer scalar indicating the desired autopilot design (See HDCON)

VTP The total airspeed in ft/sec.

c. Output Arguments

Z1, ZO k+1 and k commanded z (ψ) position of aircraft

ZD1, ZDO k+1 and k commanded $\dot{z}$ (ϕ) of aircraft

d. COMMON Blocks

None

e. Error Messages

The first pass through the subroutine causes AK1, AK2, ZDDMAX, and ZCLOSE to be printed out. If DEBUG is TRUE then ZC1, ZO, ZDG, ZDO ZDD, X1, GK1, and ZDOMAG are printed out.

f. Subroutines Employed by CLXMOD

None

g. Subroutines Employing CLXMOD

PIFLIB - DPIFS

h. Concluding Remarks

None

I. COMBINE MATRICES (CORNER)

1. PURPOSE

Subroutine CORNER inserts one matrix into another such that they have a common corner.

2. USAGE

a. Calling Sequence

CALL CORNER (A,NA,B,NB,IWHERE)

b. Input Arguments

A	Matrix packed by columns; destroyed upon return.
B.	Matrix packed by columns; not destroyed upon return.
NA,NB	Two-dimensional vectors giving number of rows and columns of respective matrices, for example: NA(1) = number of rows of A NA(2) = number of columns of A Not destroyed upon return.
IWHERE	Option for combination of the two matrices: 1 use lower left corner as common, 2 use upper right corner as common.

c. Output Arguments

A	Matrix packed by columns containing A & B combined.

d. COMMON Blocks

None

e. Error Messages

If either NA(1) < NB(1) or NA(2) < NB(2), or NA(1) x NA(2) < 1, or NB(1) x NB(2) < 1, the message "DIMENSION ERROR IN CORNER NA = _____ _____, NB = _____ _____" is printed and the program is returned to the calling point.

f. Subroutines Employed by CORNER

None

g. Subroutines Employing CORNER

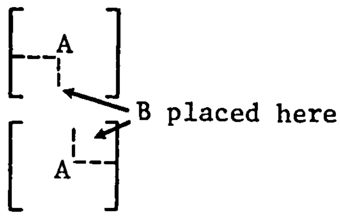
PIFLIB - PIFMODL, QRMP1, QRMHAT, PIFG, FGAERO, PIFEIG

h. Concluding Remarks

Matrix B replaces a portion of A. Therefore A must be as big or bigger than B. This is essentially an overlaying process. If IWHERE is 1, B is positioned such that the lower left corner element of A is replaced with the lower left corner of B and so forth. IWHERE of 2 indicates that the upper right corner of A and B are made equal. If B is smaller than A, one corner of B will occur part way into A. See Diagram below.

IWHERE = 1

IWHERE = 2



J. DIAGONAL PARTITION (DIAGPAR)

1. PURPOSE

To combine two matrices such that they become the top left and bottom right matrices of a quad partition matrix. The top right and bottom left matrices are set to zero in the quad partition matrix.

2. USAGE

a. Calling Sequence

CALL DIAGPAR (A,NA,B,NB,C,NC)

b. Input Arguments

A,B — Matrices packed by columns in one-dimensional arrays; not destroyed upon return.

NA,NB — Two-dimensional vectors giving the number of rows and columns of respective matrices; for example:

NA(1) = Number of rows of A

NA(2) = Number of columns of A

Not destroyed upon return.

c. Output Arguments

C — Matrix packed by columns in a one-dimensional array. Upon return A becomes the upper left hand matrix and B becomes the lower left hand matrix within the quad partition matrix C. If NB(1) = 0, then C = A.

NC — Two dimensional vector: Upon return,

NC(1) = NA(1) + NB(1)

NC(2) = NA(2) + NB(2)

unless NB(1) = 0, then

NC(1) = NA(1)

NC(2) = NA(2)

d. COMMON Blocks

None

e. Error Messages

None

f. Subroutines employed by DIAGPAR

ORACLS - NULL and EQUATE.

g. Subroutines employing DIAGPAR

PIFLIB - PIFMODL, PIFG, FGAERO, PIFEIG.

h. Concluding Remarks

$$C = \left[\begin{array}{c|c} A & 0 \\ \hline 0 & B \end{array}\right]$$

K. FORM A DIAGONAL MATRIX (DIAGPUT)

1. PURPOSE

Places elements of a vector along the diagonal of a square matrix.

2. USAGE

a. Calling Sequence

CALL DIAGPUT (A,NA,B,NB)

b. Input Arguments

A	Vector of elements to be placed on diagonal of output matrix.
NA	Number of elements in A.

c. Output Arguments

B	Matrix with diagonal elements from A packed by columns in one dimensional array, off diagonal elements are zero.
NB	Two-dimensional vector containing number of rows and columns in B. NB(1) = NA NB(2) = NA

d. COMMON blocks

None

e. Error Messages

None

f. Subroutines employed by DIAGPUT:

ORACLS - NULL

g. Subroutines employing DIAGPUT:

PIFLIB - CFWM, FGAERO, PIFEIG.

h. Concluding Remarks

Example $A = \begin{bmatrix} a, & b, & c, & d, & e \end{bmatrix}$

$$B = \begin{bmatrix} a & 0 & 0 & 0 & 0 \\ 0 & b & 0 & 0 & 0 \\ 0 & 0 & c & 0 & 0 \\ 0 & 0 & 0 & d & 0 \\ 0 & 0 & 0 & 0 & e \end{bmatrix}$$

L. DIMENSIONS (DIMSS)

1. PURPOSE

The subroutine DIMSS loads the integer dimensions into the two dimensional vectors.

2. USAGE

a. Calling Sequence

CALL DIMSS (N, M, L, NM, MM, NF, NQ, NAM, NR, NG, NH, ND, NFM, NGM, NHM, NDM, NPH, NGA, NQH, NRH, NAMH, NGK, NX, NPHC, NGKQ, NGKI, NQZ)

b. Input Arguments

N, M, L,
NM, MM — Integer scalars indicating matrix dimensions. N is the number of plant states, M is the number of plant controls, L is the number of plant commands, NM is the number of model states and MM is the number of model controls.

c. Output Arguments

NF, NQ, NAM, NR,
NH, NG, ND, NFM,
NGM, NHM, NDM,
NPH, NGA, NQH,
NRH, NAMH, NGK,
NX, NPHC, NGKQ,
NGKI, NQZ — Two dimensional integer vectors holding the number of rows and columns of the matrix shown after the letter N.

d. COMMON Blocks

None

e. Error Messages

None

f. Subroutines Employed by DIMSS

PIFLIB - PRNTITL

g. Subroutines Employing DIMSS

PIFLIB - PIFCGT

h. Concluding Remarks

None

M. DISCRETE MODEL CONSTRUCTION (DISCMOD)

1. PURPOSE

Subroutine DISCMOD transforms the continuous-time representations of the plant

$$\dot{\underline{x}} = F\underline{x} + G\underline{u} \tag{1}$$

and model

$$\dot{\underline{x}}_m = F_m\underline{x}_m + G_m\underline{u}_m \tag{2}$$

to the equivalent discrete representations

$$\underline{x}_{k+1} = DF\ \underline{x}_k + DG\ \underline{u}_k \tag{3}$$

$$\underline{x}_{m,k+1} = DFM\ \underline{x}_{m,k} + DGM\ \underline{u}_{m,k} \tag{4}$$

The controls $\underline{u}_k$ and $\underline{u}_{m,k}$ are assumed constant over the uniform sampling interval, DELT. The equations, using the plant dynamics as an example, are:

$$DF = e^{F*DELT} \tag{5}$$

$$DG = \left[\int_0^{DELT} e^{F*S}\, ds\right] * G \tag{6}$$

2. USAGE

a. Calling Sequence

CALL DISCMOD (F, NF, G, NG, FM, NFM, GM, NGM, DF, DG, DFM, DGM, W9, NW9DIM, DELT, DEBUG, ICH)

b. Input Arguments

F, G, FM,	
GM	Matrices packed by columns into one-dimensional arrays. Not destroyed upon return.
NF, NG,	
NFM, NGM	Two dimensional integer vectors holding the number of rows and columns of the matrix shown after the letter N.
W9	Work matrix packed by columns into a one-dimensional array. Dimension must be at least 2*(NF(1) ** 2 + 1)
ICH	Two-dimensional logical matrix dimensioned 10 x 15. Subroutines in PIFCGT are called depending on the TRUE or FALSE status of elements in ICH.
DEBUG	Logical scalar indicating the following: TRUE: Print out debug information FALSE: Do not print out debug information
DELT	Real scalar containing the sampling time.
NW9DIM	Integer scalar containing the maximum dimension of the W9 matrix.

c. Output Arguments

DF, DG,	
DFM, DGM	Matrices packed by columns into one dimensional arrays. If ICH (3,1) is TRUE then the matrices contain the discrete plant representations. If ICH (3,1) is FALSE then DF, DG, DFM, DGM are set equal to F, G, FM, GM.

d. COMMON Blocks

None

e. Error Messages

If DEBUG is TRUE then DFM, DGM, DF, DG are printed.

f. Subroutines Employed by DISCMOD

ORACLS - NORMS, EXPINT, UNITY, EQUATE, SCALE, NULL, ADD, LNCNT, MAXEL

PIFLIB - PRNTITL, PRNT

g. Subroutines Employing DISCMOD

PIFLIB - PIFCGT

h. Concluding Remarks

The ORACLS subroutine EXPINT cannot compute DFM if FM is the null matrix. Special logic is used in DISCMOD to circumvent the problem in EXPINT.

N. DISCRETE PIF CONTROL LAW SIMULATION (DPIFS)

1. PURPOSE

The purpose of DPIFS is to simulate the linear PIF control law tracking the output of the chosen autopilot model being designed. The output of DPIFS is a step by step print out of the simulation. The discrete equations simulated by DPIFS are as follows:

PLANT DYNAMICS

$$x_{k+1} = DF\ x_k + DG\ u_k \tag{1}$$

PLANT MEASUREMENTS

$$\underline{z}_k = HX\ \underline{x}_k + DX\ \underline{u}_k \tag{2}$$

MODEL DYNAMICS

$$x_{m,k+1} = DFM\ x_{m,k} + DGM\ u_{m,k} \tag{3}$$

OR

CLXMOD, CLYMOD, or CLXMOD is called depending on the value of ICHOSE

PIF CONTROL LAW

$$\underline{e}_k = \underline{z}_k - S_{11}\underline{x}_{m,k}, \quad \underline{y}_k = H\underline{x}_k + D\underline{u}_k, \quad \underline{y}_{m,k} = H_m\underline{x}_{m,k} + D_m\underline{u}_{m,k+1} \tag{4}$$

$$\underline{v}_k = (I - \Delta t C_4)\underline{v}_{k-1} - C_3(\underline{e}_k - \underline{e}_{k-1})$$

$$- \Delta t C_4(H\underline{x}_{k-1} + D\underline{u}_{k-1} - H_m\underline{x}_{m,k-1} - D_m\underline{u}_{m,k}) + E_1(\underline{u}_{m,k+1} - \underline{u}_{m,k}) \tag{5}$$

$$\underline{u}_{k+1} = \underline{u}_k + \Delta t\ \ \underline{v}_k + S_{21}(\underline{x}_{m,k+1} - \underline{x}_{m,k}) \tag{6}$$

RELATIONSHIP BETWEEN EQUATION VARIABLES AND DPIFS NOTATION

EQUATION VARIABLE	DPIFS NOTATION
S_{11}	S11
$I-\Delta t\ c_4$	S31
S_{21}	S21
$-C_3$	S32
$-\Delta t\ C_4$	S41
E_1	S42
$\underline{x}_{m',k+1}$	XM
$\underline{x}_{m,k}$	XM1
$\underline{x}_{m,k-1}$	XM2
$\underline{u}_{m,k+1}$	YS(1, IP+2)
$\underline{u}_k$	PU(1, IP)
$\underline{y}_k$	PY(1, IP)
$\underline{y}_{m,k}$	YSS(1, IP)
$\underline{y}_{k-1}$	YK1
$\underline{z}_k^* = S_{11}\underline{x}_{m',k}+S_{12}\underline{u}_{m',k+1}$	YX(1, IP)
$\underline{x}_k$	PX(1, IP)
$\underline{z}_k$	YXK
$\underline{e}_k$	EK1
$\underline{u}_k^* = S_{21}\underline{x}_{m',k}+S_{22}\underline{u}_{m',k+1}$	YU(1, IP)
$\underline{e}_{k-1}$	EK2
$\underline{v}_k$	VK1
$\underline{v}_{k-1}$	VK2
$\underline{y}_k$	YK
$\underline{y}_{m,k}$	YMK
$\underline{y}_{m',k-1}$	YMK1
$\underline{z}k-1$	PX(1, IP-1)

The simulation begins by setting almost all state variables to zero, the initial condition starting point. If ICHOSE = 14 (APR GS), the initial value for d is loaded in appropriate locations. If ICHOSE = 9 (APR LOCI) or ICHOSE = 15 (APR LOCR or APR LOCP), the initial value for y, and the intercept yaw angle are loaded in appropriate locations. The initial values for the commands are in UMAG(1,1) and UMAG(2,1), as discussed in HDCON. The starting time at which the command is changed from 0.0 to the UMAG value is in USTRT(1,1) and USTRT(2,1). The stopping time at which the command is returned to 0.0 is in USTP(1,1) and USTP(2,1). The simulation is from 0.0 seconds to (IMAX-2) * DELT seconds. All the data from the simulation are stored in two dimensional matrices (PX, YX, PU, YU, PY, YSL). The dimensions of these matrices limit the value of IMAX (300 for the NAVION autopilot designs).

2. USAGE

a. Calling Sequence

CALL DPIFS (DEBUG, METERS, DEBUG1, ICHOSE, ZDDMAX, ZCLOSE, ZDOMAX, ZCLMIN, ZCDMAX, VTP, LAMDA, LAMDAGS, GKDSLP, GKDICP, GKDMAX, GKDMIN)

b. Input Arguments

ZDDMAX, ZCLOSE,
ZDOMAX, AK1,
AK2, ZCLMAX,
ZCLMIN, LAMDA,
ZCDMAX, LAMDAGS,
GKDSLP, GKDICP,
GKDMAX,
GKDMIN   Autopilot parameters specified by the designer in NAMELIST.

Definitions for these variables are provided in CLXMOD, CLYMOD, and CLDMOD documentation.

DEBUG,

DEBUG1 Logical scalars indicating the following:

TRUE: Print out debug information.

FALSE: Do not print out debug information.

DEBUG is used in the subroutines CLYMOD, CLXMOD, CLDMOD. DEBUG1 is not used.

METERS If METERS is TRUE then state variables that have units of feet are converted to units of meters before print out and plotting.

ICHOSE Integer scalar indicating the desired autopilot design. (See HDCON)

c. Output Arguments

None

d. Common Blocks

DISCSV, WORKS9, MODF, PLA, DIMN1, DIMN3, DIMN4, DIMN5, SMAT, PIFG, EIGEN, LABEL, UDIR, GLAD, DISCM, HDXA

e. Error Messages

If DEBUG is TRUE then the following matrices are printed out, DFM, DGM, UMAG, USTRT, USTP.

f. Subroutines Employed by DPIFS

ORACLS - NULL, MULT, ADD, SUBT, EQUATE

PIFLIB - PRNT2, PRNTITL, CLXMOD, CLYMOD, CLDMOD

g. Subroutines Employing DPIFS

PIFLIB - PIFCGT

h. Concluding Remarks

DPIFS does not include plotting.

O. DISCRETE RICCATI EQUATION SOLUTION (DREG)

1. PURPOSE

Subroutine DREG solves the discrete Riccati equation:

$$X = PHI^T \; X \; PHI + QHAT - (GAMA^T \; X \; PHI + AMHAT^T)^T \; *$$

$$(RHAT + GAMA^T \; X \; GAMA)^{-1} \quad (GAMA^T \; X \; PHI + AMHAT^T) \qquad (1)$$

for X, then computes the feedback gain

$$GK = -\; (RHAT + GAMA^T \; X \; GAMA)^{-1} \; *$$

$$(GAMA^T \; X \; PHI + AMHAT^T) \qquad (2)$$

The solution is obtained using NEWTON'S method as discussed in ORACLS documentation of the ASYMREG subroutine. The cross weighting matrix AMHAT is eliminated from the quadratic scalar function before solving the Riccati equation using the procedure outlined in the ORACLS subroutine PREFIL. If ICH(2,7) is TRUE, the input gain GK must cause the closed-loop plant system matrix

$$PHICL = PHI + GAMA \; GK \qquad (3)$$

to be asymptotically zero. If ICH(2,7) is FALSE the subroutine attempts to find an initial starting stabilizing gain, GK, using the ORACLS subroutine DSTAB.

2. USAGE

a. Calling Sequence

CALL DREG (RHAT, NRH, AMHAT, NAM, QHAT, NQH, GAMA, NGA, X, NX, PHI,

NPH, GK, NGK, PHICL, NPHC, ICH, DEBUG, ZERO, NZERO, W9, NW9DIM, IOP)

b. Input Arguments

RHAT, AMHAT,

QHAT, GAMA,

PHI — Matrices packed by columns into one-dimensional arrays. Not destroyed upon return.

GK — Matrix packed by column into a one-dimensional array. Destroyed upon return. GK is assumed to contain a stabilizing gain if ICH(2,7) is TRUE.

W9 — Work matrix packed by columns into a one-dimensional array. Dimension must be at least 9*(NPH(1)*NPH(2) + 1)

ZERO — Work matrix packed by columns into a one-dimensional array. All elements must be zero. Dimension must be at least NPH (1) * NPH(2) + 1

ICH — Two-dimensional logical matrix dimensioned 10 x 15. Subroutines in PIFCGT are called depending on the TRUE or FALSE status of elements in ICH.

NGK, NX,

NQH, NAM, NPH,

NGA, NZERO,

NPHC — Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N. NZERO is the only vector destroyed upon return.

DEBUG — Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug information

DELT — Real scalar containing the sampling time.

NW9DIM Integer scalar containing the maximum dimension of the W9 matrix.

IOP(1)=0 Do not print results. Otherwise print input data, QHAT and final values of GK and X.

IOP(2)=0 Do not print at each iteration. Otherwise regardless of printing specified by IOP(1), print iteration count and value of X.

IOP(3) Not Required.

IOP(4)=0 Do not compute the Riccati equation residual. Otherwise compute the residual and print.

IOP(5)=0 Compute but do not print the eigenvalues of X, the matrix (PHI - GAMA GK) and the eigenvalues of (PHI - GAMA GK). Otherwise print these data after computation.

c. Output Arguments

X, GK,

PHICL Matrices packed by columns into one-dimensional arrays containing respectively the discrete Riccati equation solution, the discrete feedback gain, and the discrete closed-loop system matrix.

NX, NGK,

NPHC Two dimensional integer vectors holding the number of rows and columns of the matrix shown after the letter N.

d. COMMON Blocks

None

e. Error Messages

If the workspace in the W9 matrix is too small the message "WORK SPACE IS TOO SMALL FOR CALCULATION". WORK SPACE = ________. WC = _______.

The solution to the RICCATI equation (X) and the gain matrix (GK) are always printed out.

f. Subroutines Employed by DREG

ORACLS - MULT, SUBT, TESTSTA, SCALE, RICTNWT, TRANP, ADD, EQUATE, EIGEN,
JUXTC, DSTAB, UNITY, NULL, ASYMREG,

PIFLIB - PRNTITL, PRNT

g. Subroutines Employing DREG

PFICGT

h. Concluding Remarks

DSTAB fails consistently in computing a stabilizing gain. An alternate method for computing the starting gain is performed by PIFCGT.

P. SUBSET OF A MATRIX (EXTR)

1. PURPOSE

Subroutine EXTR extracts a matrix from within another matrix.

2. USAGE

a. Calling Sequence

CALL EXTR(A,NA,B,NB,I,J)

b. Input Arguments

A Matrix packed by columns in one dimensional array. Not destroyed upon return.

NA 2 element vector giving number of rows and columns respectively of A. Not destroyed upon return.

NB 2 element vector giving number of rows and columns to be extracted into B. Not destroyed upon return.

I & J Define upper left hand corner position of B as a row and column position in A.

c. Output Arguments

B Matrix that is extracted. Packed column wise into one dimensional array.

d. COMMON Blocks

None

e. Error Messages

If either NA(1) < NB(1), or NA(2) < NB(2), or NA(1) x NA(2) < 1, or NB(1) x NB(2) < 1, the message is "DIMENSION ERROR IN CORNER NA = _____ _____, NB = _____ _____" is printed and the program is returned to the calling point.

f. Subroutines employed by EXTR

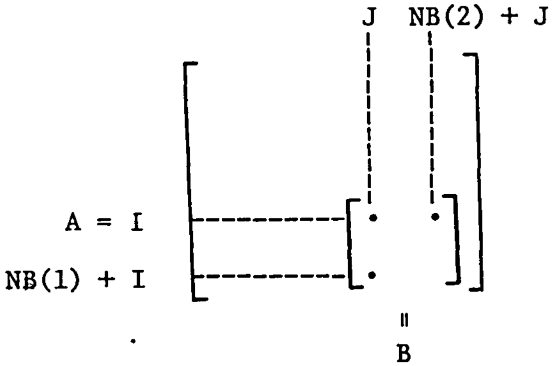
None

g. Subroutines employing EXTR

PIFLIB - CFWM, PIFPLC, PIFG, HADAC.

h. Concluding Remarks

This routine is useful for extracting relevant sub-matrices from within a partitioned matrix as shown below.



Q. F AND G CONSTRUCTION FROM AERODYNAMIC COEFFICIENTS (FGAERO)

1. PURPOSE

Subroutine FGAREO accepts nondimensional aerodynamic coefficients, trim conditions, and a desired velocity magnitude and constructs the aircraft perturbation model matrices F and G at the trim condition specified by the desired aircraft velocity, where

$$\Delta\dot{\underline{x}} = F\ \Delta\underline{x} + G\ \Delta\underline{u} \tag{1}$$

A derivation of the perturbation aircraft dynamics based on the non-linear dynamics is summarized in Ref. 1. Information on the notation used in the following discussion can be found in Ref. 1, Appéndix A. The F and G matrices are partitioned as follows:

$$\begin{bmatrix} \Delta\dot{\underline{v}}_B \\ \Delta\dot{\underline{\omega}}_B \\ \Delta\dot{\underline{\nu}}_B \\ \Delta\dot{\underline{x}}_B \end{bmatrix} = \begin{bmatrix} A11 & A12 & A13 & 0 \\ A21 & A22 & 0 & 0 \\ 0 & A32 & A33 & 0 \\ A41 & 0 & A43 & 0 \end{bmatrix} \begin{bmatrix} \Delta\dot{\underline{v}}_B \\ \Delta\dot{\underline{\omega}}_B \\ \Delta\dot{\underline{\nu}}_B \\ \Delta\dot{\underline{x}}_B \end{bmatrix}$$

$$+ \begin{bmatrix} B11 \\ B21 \\ 0 \\ 0 \end{bmatrix} \Delta\underline{u} + \begin{bmatrix} F1 \\ F2 \\ 0 \\ 0 \end{bmatrix} \Delta\dot{\underline{\nu}}_w \tag{2}$$

The state and control vectors are

$$[\Delta\underline{v}_B^T\ \ \Delta\underline{\omega}_B^T\ \ \Delta\underline{\nu}_B^T\ \ \Delta\underline{x}_E^T] = [\Delta u\ \Delta v\ \Delta w\ \ \ \Delta p\ \Delta q\ \Delta r\ \ \ \Delta\phi\ \Delta\theta\ \Delta\psi\ \ \ \Delta x\ \Delta y\ \Delta z] \tag{3}$$

$$\Delta\underline{u}^T = [\Delta\delta_T \quad \Delta\delta_e \quad \Delta\delta_a \quad \Delta\delta_r] \tag{4}$$

$$\Delta\underline{\dot{\nu}}_w = [\Delta\dot{\beta} \; \Delta\dot{\alpha}] \tag{5}$$

The effect of $\Delta\underline{\dot{\nu}}_w$ is included after the above partitions have been computed. The partitioned matrices in Eq. 2 satisfy the following:

$$\text{A11} = \text{FBV} - \text{WBI} + \text{FBAB*AU} \tag{6}$$

$$\text{A12} = \text{VBT} + \text{FBW} \tag{7}$$

$$\text{A13} = \text{NBV} \tag{8}$$

$$\text{A21} = \text{IB}^{-1}\text{*(GBV} + \text{GBAB*AU)} \tag{9}$$

$$\text{A22} = \text{IB}^{-1}\text{*GBW} - \text{IBIB} \tag{10}$$

$$\text{A32} = \text{LB}^{-1} \tag{11}$$

$$\text{A33} = \text{LBLB} \tag{12}$$

$$\text{A41} = (\text{HBI})^T = \text{HIB} \tag{13}$$

$$\text{A43} = -\widetilde{(\text{HIB*VB})}\text{*HIB*LB} \tag{14}$$

$$\text{F1} = 0 \tag{15}$$

$$\text{F2} = \text{IB}^{-1}\text{*GBABD} \tag{16}$$

$$\text{B11} = \text{FBU} \tag{17}$$

$$\text{B21} = \text{IB}^{-1}\text{*GBU} \tag{18}$$

$$\text{LB} = \begin{bmatrix} 1 & 0 & -\sin(\text{THETAO}) \\ 0 & \cos(\text{PHIO}) & \sin(\text{PHIO})\cos(\text{THETAO}) \\ 0 & -\sin(\text{PHIO}) & \cos(\text{PHIO})\cos(\text{THETAO}) \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \text{LB} \begin{bmatrix} \text{PO} \\ \text{QO} \\ \text{RO} \end{bmatrix} = \text{LB * WB} \tag{20}$$

$$\text{LBLB} = \begin{bmatrix} \dot{\theta}\ \text{TAN}\left(\text{THETAO}\right) & \dot{\psi}\ \text{SEC}\left(\text{THETAO}\right) & 0 \\ -\dot{\psi}\ \text{COS}\left(\text{THETAO}\right) & 0 & 0 \\ \dot{\theta}\ \text{SEC}\left(\text{THETAO}\right) & \dot{\psi}\ \text{TAN}\left(\text{THETAO}\right) & 0 \end{bmatrix} \tag{21}$$

$$\text{VB} = \begin{bmatrix} \text{UO} \\ \text{VO} \\ \text{WO} \end{bmatrix} \tag{22}$$

$$\text{VBT} = \begin{bmatrix} 0 & -\text{WO} & \text{VO} \\ \text{WO} & 0 & -\text{UO} \\ -\text{VO} & \text{UO} & 0 \end{bmatrix} = \widetilde{\text{VB}} \tag{23}$$

$$\text{WB} = \begin{bmatrix} \text{PO} \\ \text{QO} \\ \text{RO} \end{bmatrix} \tag{24}$$

$$\text{WBI} = \begin{bmatrix} 0 & -\text{RO} & \text{QO} \\ \text{RO} & 0 & -\text{PO} \\ -\text{QO} & \text{PO} & 0 \end{bmatrix} = \widetilde{\text{WB}} \tag{25}$$

$$\text{HBI} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\left(\text{PHIO}\right) & \sin\left(\text{PHIO}\right) \\ 0 & -\sin\left(\text{PHIO}\right) & \cos\left(\text{PHIO}\right) \end{bmatrix} * \begin{bmatrix} \cos\left(\text{THETAO}\right) & 0 & \sin\left(\text{THETAO}\right) \\ 0 & 1 & 0 \\ \sin\left(\text{THETAO}\right) & 0 & \cos\left(\text{THETAO}\right) \end{bmatrix}$$

$$* \begin{bmatrix} \cos \text{PSIO} & \sin \text{PSIO} & 0 \\ -\sin \text{PSIO} & \cos \text{PSIO} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{26}$$

$$\underline{\text{GG}} = \begin{bmatrix} \text{G5} \\ \text{G6} \\ \text{G7} \end{bmatrix} = \text{HBI} * \begin{bmatrix} 0 \\ 0 \\ \text{GR} \end{bmatrix} \tag{27}$$

$$GB = \begin{bmatrix} 0 & -G7 & G6 \\ G7 & 0 & -G5 \\ -G6 & G5 & 0 \end{bmatrix} = \widetilde{GG} \tag{28}$$

(The subroutine CROSSP forms the cross produce equivalent matrix, GB, from the vector $\underline{GG}$)

$$IB = \begin{bmatrix} IX & 0 & 0 \\ 0 & IY & 0 \\ 0 & 0 & IZ \end{bmatrix} \tag{29}$$

$$IBIB = IB^{-1} * [WBI*IB - \widetilde{(IB*WB)}] \tag{30}$$

$$S = RHO * A * 0.5 \tag{31}$$

$$AMASSI = GR/WT \tag{32}$$

$$FBV = 2*S* \begin{bmatrix} CXO*UO & CXO*VO & CXO*WO \\ CYO*UO & CYO*VO & CYO*WO \\ CZO*UO & CZO*VO & CZO*WO \end{bmatrix} \tag{33}$$

$$FBW = 0.5*S*VTO* \begin{bmatrix} B*CXP & C*CXQ & B*CXR \\ B*CYP & C*CYQ & B*CYR \\ B*CZP & C*CZQ & B*CZR \end{bmatrix} \tag{34}$$

$$FBAB = S*VTO**2 * \begin{bmatrix} CXB & CXA \\ CYB & CYA \\ CZB & CZA \end{bmatrix} \tag{35}$$

$$FBU = S*VTO**2 * \begin{bmatrix} CXDT & CXDE & CXDA & CXDR \\ CYDT & CYDE & CYDA & CYDR \\ CZDT & CZDE & CZDA & CZDR \end{bmatrix} \tag{36}$$

$$GBV = 2*S * \begin{bmatrix} B*CLO*UO & B*CLO*VO & B*CLO*WO \\ C*CMO*UO & C*CMO*VO & C*CLO*WO \\ B*CNO*UO & B*CNO*VO & B*CNO*WO \end{bmatrix} \tag{37}$$

$$GBAB = S*VTO**2 * \begin{bmatrix} B*CLB & B*CLA \\ C*CMB & C*CMA \\ B*CNB & B*CNA \end{bmatrix} \tag{38}$$

$$GBW = 0.5 *S*VTO * \begin{bmatrix} B**2*CLP & B*C*CLQ & B*B*CLR \\ B*C*CMP & C*C*CMQ & B*C*CMR \\ B**2*CNP & B*C*CNQ & B*B*CNR \end{bmatrix} \tag{39}$$

$$GBU = S*VTO**2 * \begin{bmatrix} B*CLDT & B*CLDE & B*CLDA & B*CLDR \\ C*CMDT & C*CMDE & C*CMDA & C*CMDR \\ B*CNDT & B*CNDE & B*CNDA & B*CNDR \end{bmatrix} \tag{40}$$

$$GBABD = 0.5*S*VTO * \begin{bmatrix} B*B*CLBDOT & C*C*CLADOT \\ B*C*CMBDOT & C*C*CMADOT \\ B*B*CNBDOT & B*C*CNADOT \end{bmatrix} \tag{41}$$

$$AU = \begin{bmatrix} 0 & \frac{1.0}{VTO} & 0 \\ \frac{WO}{UO**2+WO**2} & 0 & \frac{UO}{UO**2+WO**2} \end{bmatrix} \tag{42}$$

$$ABV = \begin{bmatrix} 0 & \frac{1}{VTO} & 0 \\ 0 & 0 & \frac{1}{VTO} \end{bmatrix} \tag{43}$$

$$Z = I - ABV * F1 \tag{44}$$

$$G1 = Z^{-1} \; [ABV*A11 \quad ABV*A12 \quad ABV*A13 \quad 0]$$

$$G2 = Z^{-1} * ABV * B1 \tag{45}$$

The effect $\Delta\dot{\underline{\nu}}_w$ is included to produce the final form for F and G

$$\Delta\dot{\underline{\nu}}_w = G1\ \ \underline{x} + G2\ \ \underline{u} \tag{46}$$

$$F = F + \begin{bmatrix} 0 \\ F2 \\ 0 \\ 0 \end{bmatrix} * G1 \tag{47}$$

$$G = G + \begin{bmatrix} 0 \\ F2 \\ 0 \\ 0 \end{bmatrix} * G2 \tag{48}$$

After F and G have been constructed the states are reordered as follows:

$$\Delta\underline{x}^T = [\Delta u\ \Delta w\ \Delta q\ \Delta\theta\ \Delta z\ \Delta x\ \Delta v\ \Delta r\ \Delta p\ \Delta\phi\ \Delta\psi\ \Delta y] \tag{49}$$

completing the linear time-invariant aircraft model construction. If ICH(1,5) is TRUE, the units are changed from radians to degrees.

2. USAGE

a. Calling Sequence

CALL FGAERO (DEBUG, ICH, HBI, NHBI, LB, NLB)

b. Input Arguments

DEBUG — Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug information

ICH — Two-dimensional logical matrix dimensioned 10 x 15. Subroutines in PIFCGT are called depending on the TRUE or FALSE status of elements in ICH.

c. Output Arguments

HBI, LB — Matrices packed by columns into one dimensional arrays.

NHBI, NLB — Two dimensional integer vector holding the number of rows and columns of the matrix after the letter N.

d. COMMON Blocks

PLA, GLAD, DIMN4, AERO, WORKS9, HDXA

e. Error Messages

If the LB matrix is not invertible the message "LB IS A SINGULAR MATRIX" is printed and the program stops. If the Z matrix is not invertible the message "Z IS A SINGULAR MATRIX" is printed and the program stops. If DEBUG is TRUE the following statements and matrices are printed during the course of program execution

*** s = ________

*** AMASSI = ________

LB, HBI, GB, NBV, LB^{-1}(LB1), LBLB, IB, IB^{-1}(IB1), IBIB, FBV, FBW, FBAB, FBU, GBV, GBAB, GBW, GBU, GBBD, A11, A12, A13.

A1 The F matrix in Eq. 2, before including the effect of $\Delta\underline{\dot{\nu}}_w$ and the position states $\Delta\underline{x}_E$

B11, B21,

B31, B1 The G matrix in Eq. 2, before including the effect of $\Delta\underline{\nu}_w$ and the position states $\Delta\underline{x}_E$

F11, F21, F31, F1, Z, Z^{-1}(ZINV), G1, G2,

FORD The A1 matrix including the effect of $\Delta\underline{\nu}_w$

GHIA The B1 matrix including the effect of $\Delta\underline{\nu}_w$

If ICH(1,5) is TRUE then the following are printed

FDEG The F matrix in Eq. 1 using degree units

GDEG The G matrix in Eq. 1 using degree units

f. Subroutines Employed by FGAERO

ORACLS - EQUATE, MULT, UNITY, GELIM, DETFAC, GAUSEL, SUBT, SCALE, ADD, JUXTC, JUXTR

PIFLIB - PRNTITL, PRNT, MOVEALL, CORNER, DIAGPAR, CROSSP, R3MAT

g. Subroutines Employing FGAERO

PIFLIB - PIFCGT

h. Concluding Remarks

None

R. EIGENVALUES/EIGENVECTORS OF CLOSED-LOOP PLANTS USING CSQZ (FREEF)

1. PURPOSE

The purpose of FREEF is to compute eigenvalues, λ , and eigenvectors, $\underline{x}$, of the matrix, F, where

$$F\underline{x} = \lambda\underline{x} \quad , \quad \lambda = R + jI \tag{1a, b}$$

FREEF assumes F is a closed-loop plant matrix

$$F = A + BK \tag{2}$$

so that eigenvalues of F are interpreted as indications of closed-loop stability. FREEF uses the NASA Langley FTNMLIB subroutine CSQZ. CSQZ uses Householder transformations and a combination of the double shift QZ iteration and the single shift implicit QZ iteration. Consult Section 2h for further remarks.

2. USAGE

a. Calling Sequence

CALL FREEF (F, ER, EI, V, WK, IFREE, N)

b. Input Arguments

F	Matrix packed by columns into one-dimensional array. Destroyed upon return.
EFREE	Integer scalar for eigenvalue computations 0 - No calculation in FREEF 1 - Compute and print eigenvalues 2 - Compute and print eigenvalues and eigenvectors
N	Integer scalar indicating row dimension of the square matrix F.

WK Work matrix of dimension at least 2*(N**2+1)

c. Output Arguments

ER Real vector containing the real part of the eigenvalues in ascending order of magnitude

EI Real vector containing the imaginary part of the eigenvalues in ascending order of magnitude

WK Real matrix containing the eigenvectors packed by columns into a one-dimensioned array. A real eigenvector is placed in one column. A complex eigenvector is placed in two columns. The real part is placed in the first column. The imaginary part of the positive complex eigenvector pair is placed in the second column. The arrangement is in ascending order of magnitude.

d. COMMON Blocks

None

e. Error Messages

After CSQZ is called, the following is printed:

OUTPUT OF CSQZ IN FREE

EPSA = _____ EPSB = _____ IERR = _____

Eigenvalues are calculated as $(\alpha_1+j\alpha_2)/\beta$. EPSA is the inaccuracy level of $|\alpha_1+j\alpha_2|$. (Digits below this level are known to be inaccurate). EPSB is the inaccuracy level of β. IERR is an error code.

IERR = 0 normal return.

= I Ith eigenvalue ratio has not been determined after 30 iterations.

If β is zero for eigenvalue I the following is printed:

EIGENVALUE PROBLEM IN CSQZ INFREE: BETA IS ZERO I = __.

Additional information for each eigenvalue is printed with the following banners:

EIGENVALUES	The values for the eigenvalues R and jI
TIME CONSTANT	-1.0/R - Time to reach 63.212% of a step command
50 PERCENT SET TIME	0.6931472*(-1.0/R) - Time to reach 50% of a step command
10 PERCENT SET TIME	2.302582*(-1.0/R) - Time to reach 90% of a step command
2 PERCENT SET TIME	3.9120230*(-1.0/R) - Time to reach 98% of a step command
UNDAMPED NAT FREQ	(R**2+I**2) - Undamped natural frequency of a complex eigenvalue
RATIO	-R/(R**2+I**2) - Damping ratio of a complex eigenvalue
RISE TIME	0.5*(6.28318531)/|I| - TAN^{-1} (|I|/R)/|I| - time for the step response of a complex eigenvalue to cross the step command the first time.
PERIOD	6.28318531/|I| - Time for a complex eigenvalue to complete one oscillation.
OVERSHOOT	100.0* $e^{(0.5*R*6.28318531/\|I\|)}$ - The overshoot of a stable complex eigenvalue in response to a step input.

f. Subroutines Employed by FREEF

ORACLS - UNITY

PIFLIB - RESPON

FTNMLIB - CSQZ

g. Subroutines Employing FREEF

None

h. Concluding Remarks

A user with access to the subroutine CSQZ should replace the call to FREEO with a call to FREEF in the following subroutines - PIFCGT, PIFEIG, PIFPLC, PIFFRE. A user without access to CSQZ should remove the subroutine FREEF from the software package.

S. EIGENVALUES/EIGENVECTORS OF CLOSED-LOOP PLANTS USING EIGEN (FREEO)

1. PURPOSE

The purpose of FREEO is to compute eigenvalues, λ, and eigenvectors, $\underline{x}$, of the matrix, F, where

$$F\underline{x} = \lambda\underline{x} \quad , \lambda = R + jI \tag{1a, b}$$

FREEO assumes F is a closed-loop plant matrix

$$F = A + BK \tag{2}$$

so that eigenvalues of F are interpreted as indications of closed-loop stability. FREEO uses the ORACLS subroutine EIGEN. EIGEN balances the matrix and the eigenvalues are found by stabilized elementary similarity transformations and the double shift QR algorithm.

2. USAGE

a. Calling Sequence

CALL FREEO (F, ER, EI, V, WK, IFREE, N)

b. Input Arguments

F — Matrix packed by columns into one-dimensional array; destroyed upon return.

IFREE — Integer scalar for eigenvalue computations

0 - No calculation in FREEF

1 - Compute and print eigenvalues

2 - Compute and print eigenvalues and eigenvectors

N — Integer scalar indicating row dimension of the square matrix F

WK — Work matrix of dimension at least 2*(N**2+1)

c. Output Arguments

ER Real vector containing the real part of the eigenvalues in ascending order of magnitude

EI Real vector containing the imaginary part of the eigenvalues in ascending order of magnitude

WK Real matrix containing the eigenvectors packed by columns into a one-dimensional array. A real eigenvector is placed in one column. A complex eigenvector is placed in two columns. The real part is placed in the first column. The imaginary part of the positive complex eigenvector pair is placed in the second column. The arrangement is in ascending order of magnitude.

d. COMMON Blocks

None

e. Error Messages

After EIGEN is called and if IERR (defined in EIGEN documentation) is nonzero the value of IERR is printed out. Additional information for each eigenvalue is printed with the following banners:

EIGENVALUES	The values for the eigenvalues R + jI
TIME CONSTANT	-1.0/R - Time to reach 63.212% of a step command
50 PERCENT SET TIME	0.6931472*(-1.0/R) - Time to reach 50% of a step command
10 PERCENT SET TIME	2.302582*(-1.0/R) - Time to reach 90% of a step command
2 PERCENT SET TIME	3.9120230*(01.0/R) - Time to reach 98% of a step command
UNDAMPED NAT FREQ	(R**2+I**2) - Undamped natural frequency of a complex eigenvalue

RATIO	-R/(R**2+I**2) - Damping ratio of a complex eigenvalue
RISE TIME	$0.5*(6.28318531/\|I\|) - TAN^{-1}(\|I\|/R)/\|I\|$ - Time for the step response of a complex eigenvalue to cross the step command the first time.
PERIOD	$6.28318531/\|I\|$ - Time for a complex eigenvalue to complete one oscillation.
OVERSHOOT	$100.0*e^{(0.5*R*6.28318531/\|I\|)}$ - The overshoot of a stable complex eigenvalue in response to a step input.

f. Subroutines Employed by FREEF

ORACLS - UNITY EIGEN

PIFLIB - RESPON

g. Subroutines Employing FREEF

PIFCGT, PIFEIG, PIFPLC, PIFFRE

h. Concluding Remarks

None

T. ACCELERATION AND VELOCITY VECTOR OBSERVATION MATRIX CONSTRUCTION (HADAC)

1. PURPOSE

HADAC constructs the 3x13 and 3x4 observation matrices HAC and DAC and the 3x12 observation matrix HAB. HAC and DAC initially are the body-mounted perturbation accelerometer observation matrices and satisfy

$$\Delta \underline{a}_B = \text{HAC } \Delta \underline{x} + \text{DAC } \Delta \underline{u} \tag{1}$$

where $\Delta \underline{a}_B$ is the perturbation body-axis accelerometer output,

$$\Delta \underline{a}_B^T = [\Delta a_x \quad \Delta a_y \quad \Delta a_z] \tag{2}$$

$\Delta \underline{x}$ is the perturbation aircraft state vector

$$\Delta \underline{x}^T = [\Delta u \ \Delta w \ \Delta v \ \Delta p \ \Delta q \ \Delta r \ \Delta\phi \ \Delta\theta \ \Delta\psi \ \Delta x \ \Delta y \ \Delta z] \tag{3}$$

$\Delta \underline{u}$ is the perturbation aircraft control vector

$$\Delta \underline{u}^T = [\Delta\delta_T \quad \Delta\delta_e \quad \Delta\delta_a \quad \Delta\delta_r] \tag{4}$$

The units must be feet and radians.

HAB is the perturbation observation matrix for total velocity, ., ΔV , sideslip, $\Delta\beta$ and angle of attack, $\Delta\alpha$, where

$$\Delta \underline{\eta} = \text{HAB } \Delta \underline{x} \tag{5}$$

and

$$\Delta \underline{\eta} = \begin{bmatrix} \Delta V \\ \Delta\beta \\ \Delta\alpha \end{bmatrix} \tag{6}$$

The construction of HAC and DAC proceeds by first partitioning the 12x12 plant system matrix, F, and 12x4 control matrix, G, as follows

$$F = \begin{bmatrix} FXX & FXW & FXV & 0 \\ FWX & FWW & FWV & 0 \\ FVX & FVW & FVV & 0 \\ FPX & FPW & FPV & 0 \end{bmatrix} \tag{7}$$

$$G = \begin{bmatrix} GXU \\ GWU \\ GVU \\ 0 \end{bmatrix} \tag{8}$$

The expressions for HAC, DAC, and HAB become (see Ref.1 for a derivation)

$$HAC = [H_1 \quad H_2 \quad H_3 \quad 0] \tag{9}$$

$$H_1 = FXX + TW - TSB * FWX \tag{10}$$

$$H_2 = T(TXB*W) - TW * TXB - TVB + FXW - TXB * FWW \tag{11}$$

$$H_3 = FXV - TXB * FWV - GB * LB \tag{12}$$

where

$$LB = \begin{bmatrix} 1.0 & 0 & -\sin\ (THETAO) \\ 0 & \cos\ (PHIO) & \sin\ (PHIO) * \cos\ (THFTAO) \\ 0 & -\sin\ (PHIO) & \cos\ (PHIO) * cso\ (THETAO) \end{bmatrix} \tag{13}$$

$$HBI = HIB^T \tag{14}$$

$$HIB = T_1 * T_2 * T_3 \tag{15}$$

$$T_1 = \begin{bmatrix} \cos\ (PSIO) & -\sin\ (PSIO) & 0 \\ \sin\ (PISO) & \cos\ (PISO) & 0 \\ 0 & 0 & 1.0 \end{bmatrix} \tag{16}$$

$$T_2 = \begin{bmatrix} \cos\ (PHIO) & 0 & \sin\ (PHIO) \\ 0 & 1.0 & 0 \\ -\sin\ (PHIO) & 0 & \cos\ (PHIO) \end{bmatrix} \tag{17}$$

$$T_3 = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & \cos\ (\text{THETAO}) & -\sin\ (\text{THETAO}) \\ 0 & \sin\ (\text{THETAO}) & \cos\ (\text{THETAO}) \end{bmatrix} \tag{18}$$

$$\text{TW} = \begin{bmatrix} 0 & -\text{RO} & \text{QO} \\ \text{RO} & 0 & -\text{PO} \\ -\text{QO} & \text{PO} & 0 \end{bmatrix} \tag{19}$$

$$\text{TXB} = \begin{bmatrix} 0 & -zs & xs \\ zs & 0 & -ys \\ -xs & -ys & 0 \end{bmatrix} \tag{20}$$

$$\text{VBT} = \begin{bmatrix} 0 & -\text{WO} & \text{UO} \\ \text{WO} & 0 & -\text{VO} \\ -\text{UO} & \text{VO} & 0 \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \text{HBI} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{22}$$

$$\text{GB} = \begin{bmatrix} 0 & -g_3 & g_1 \\ g_3 & 0 & -g_2 \\ -g_1 & g_2 & 0 \end{bmatrix} \tag{23}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \text{TXB} * \begin{bmatrix} \text{PO} \\ \text{QO} \\ \text{RO} \end{bmatrix} \tag{24}$$

$$\text{T(TXB*W)} = \begin{bmatrix} 0 & -x_3 & x_1 \\ x_3 & 0 & -x_2 \\ -x_1 & x_2 & 0 \end{bmatrix} \tag{25}$$

$$DAC = [GXU - TXB * GWU] \tag{26}$$

where

$$HAB = [JW * HWB \quad 0 \quad 0 \quad 0] \tag{27}$$

$$JW = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & \frac{1.0*DPR}{V_{T_o}} & 0 \\ 0 & 0 & \frac{1.0*DPR}{(V_{T_o}\cos\beta_o)} \\ & & 0 \end{bmatrix} \tag{28}$$

$$HWB = HBW^T \tag{29}$$

$$HBW = \begin{bmatrix} \cos\beta_o & 0 & -\sin\beta_o \\ 0 & 1.0 & 0 \\ \sin\beta_o & 0 & \cos\beta_o \end{bmatrix} \begin{bmatrix} \cos\beta_o & \sin\beta_o & 0 \\ -\sin\beta_o & \cos\beta_o & 0 \\ 0 & 0 & 1.0 \end{bmatrix} \tag{30}$$

$$DPR = 180.0/\pi \tag{31}$$

$$V_{T_o} = \sqrt{UO^2 + VO^2 + WO^2} \tag{32}$$

$$\beta_o = \sin^{-1}(VO/V_{T_o}) \tag{33}$$

$$\alpha_o = \tan^{-1}(WO/UO) \tag{34}$$

If ICH(2,12) is TRUE, Δa_x observation is replaced with velocity magnitude, ΔV, observation in HAC DAC by replacing the first row in HAC with the first row in HAB and zeroing out the first row in DAC. In a similiar manner if ICH(2,13) is TRUE, Δa_y observation is replaced with sideslipe observation, $\Delta\beta$, and if ICH(2,14) is TRUE, Δa_z observation is replaced with angle-of-attack observation, $\Delta\alpha$.

2. USAGE

a. Calling Sequence

CALL HADAC (LB, NLB, VBT, NVBT, GB, NGB, DEBUG, F, NF, G, NG, XS, YS, ZS, PO, QO, RO UO, VO, WO, HAC, DAC, ICH, NGAC, NDAC, HAB, NHAB)

b. Input Arguments

LB, GB, F,
G, VBT — Matrices packed by columns into one-dimensional arrays; not destroyed upon return. These matrices are constructed in FGAERO

NLB, NVBT,
NGB, NF, NG — Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N.

XS, YS,
ZS — Real scalars containing the x, y, and z body axis position of the accelerometers with reference to the aircraft center of gravity.

PO, QO, RO,
UO, VO, WO — Real scalar trim conditions for roll rate, pitch rate, yaw rate, body-axis forward velocity, lateral velocity and vertical velocity, respectively.

ICH — Two-dimensional logical matrix dimensioned 10x15.

Subroutines in PIFCGT are called depending on the TRUE or FALSE status of elements in ICH.

c. Output Arguments

HAC, DAC,

HAB Matrices packed by columns into one-dimensional arrays

NHAC, NDAC,

NHAB Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N.

d. COMMON Blocks

WORKS9

e. Error Messages

If DEBUG if TRUE the following matrices are printed our during the course of program execution: FXX, GXU, JW*HWB(JHWB), HAB, HAC, DAC.

f. Subroutines Employed by HADAC

ORACLS - NULL, MULT, SUBT, ADD, JUXTC, TRANP, UNITY

PIFLIB - CROSSP, EXTR, PRNT, PRNTITL

g. Subroutines Employing Hadac

PIFLIB - FGAERO

h. Concluding Remarks

None

U. FM, GM, HM, DM, H AND D MATRIX CONSTRUCTION (HDCON)

1. PURPOSE

The purpose of HDCON is to construct the FM, GM, HM, DM, H and D matrices, to set up the XR, UR, YR, XMR, UMR logical vectors used by the REMAT subroutine, and to modify the XR, UR, YR, and UMR vectors as needed to reflect changes from initial specifications in AEROD (BLOCK DATA). The program has the necessary information for each command system that is chosen by the designer using ICHOSE. For each value of ICHOSE supported by the program, (ICHOSE - 4 is available but not supported), the values for elements FM, GM, HM, H and D are shown in Ref. 1, Appendix C. The program constructs the matrices assuming the aircraft states and controls have the following order,

$$\Delta\underline{x}^T = [\Delta u \ \Delta w \ \Delta q \ \Delta\theta \ \Delta x \ \Delta z \ \Delta v \ \Delta r \ \Delta p \ \Delta\phi \ \Delta\psi \ \Delta y] \tag{1}$$

$$\Delta\underline{u}^T = [\Delta\delta_T \ \Delta\delta_e \ \Delta\delta_a \ \Delta\delta_r] \tag{2}$$

The order for $\Delta\underline{y}$, $\Delta\underline{y}_m$, $\Delta\underline{x}_m$, and $\underline{u}_m$ are the same as the order in Ref. 1.

After constructing the matrices, the states and controls which are to be eliminated are identified by setting appropriate elements in XR, UR and YR to FALSE. The order for XR and UR are the same as Eqs. 1 and 2. The first two elements in YR are assumed to be longitudinal. The latter two elements in YR are assumed to be lateral.

The weighting elements for the integrator states, (Z1Q, Z2Q, Z3Q, Z4Q), have the same order as YR. Integrator states are eliminated in REMAT using the YR logical vector. The values for NM, the number of command model controls are specified in HDCON for each ICHOSE value.

The values for L, N, and M are always 4, 12 and 4, respectively. REMAT changes these values depending on XR, UR, and YR. The plotting labels in YL must be changed to reflect different command systems. The first two rows in YL are assumed to be longitudinal. The latter two rows in YL are assumed to be lateral. Rows in YL are eliminated in REMAT using the YR logical vector.

2. USAGE

a. Calling Sequence

CALL HDCON (H, NH, D, ND, FM, NFM, GM, NGM, HM, NHM, DM, NDM, ICHOSE, W9, NW9DIM, UO, WO, QO, RO, VTO, THETAO, PHIO, DEBUG, METERS, XR, UR, YR, XMR, UMR, ICH, HAC, NHAC, DAC, NDAC, GR, N, M, L, NM, MM, XL, YL, GKPSI, GKY, F, LAMDA, ZETA)

b. Input Arguments

F — The 12 x 12 aircraft system matrix packed by columns in a one-dimensional array. The F matrix is constructed in FGAERO. The F matrix is altered if ICHOSE = 14.

ICH — Two-dimensional logical matrix dimensioned 10 x 15. Subroutines in PIFCGT are called depending on the TRUE or FALSE status of elements in ICH.

W9 — Work matrix packed by columns into a one-dimensional array. Dimension must be at least 7*(n + m + 1).

DEBUG — Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug information.

NW9DIM — Integer scalar containing the maximum dimension of the W9 matrix.

UO, WO, QO,

RO, THETAO,

LAMDA, ZETA,

VTO, PHIO	Trim values for the states. The trim values are obtained from linear interpolation in INTERP. LAMDA is the flight path angle in degrees (LAMDA = 0.0 deg usually). ZETA is the ground relative heading angle in deg (ZETA = 0.0 usually).

ICHOSE	Integer scalar indicating the desired autopilot designs as follows:

ICHOSE	DESCRIPTION
1-BETA HOLD	Beta and roll angle constant command system for lateral dynamics. Beta and roll angle perturbation commands for DPIFS simulation are in UMAG(1,1) and UMAG(2,1), respectively.
6-HDG SEL	Yaw angle outer loop turning command system using the ROLL HOLD command system as the inner loop structure. Yaw angle and rudder position perturbation commands for DPIFS simulation are in UMAG(1,1) and UMAG(2,1).
7-ALT SEL	Altitude select command system for longitudinal dynamics. Altitude perturbation command is in UMAG(1,1).
9-APR LOCI	Localizer tracking model for lateral dynamics. Perturbation lateral position offset is in UMAG(1,1). Yaw intercept angle is in UMAG(2,1). USTRT(1,1) and USTRT(2,1) must be at 0.0. USTP(1,1) and USTP(2,1) must be at desired final time.

12-ROLL HOLD	Roll angle and rudder angle constant command system. Roll and rudder angle perturbation commands are in UMAG(1,1) and UMAG(2,1).
14-APR GS	Glideslope position error (d) tracking command system. UMAG(1,1) contains the d position offset at beginning of simulation. USTRT(1,1) should be 0.0. z dynamics in F and G are replaced with d dynamics in HDCON.
15-APR LOCR	Lateral position (y) localizer outer loop command system using the HDG SEL command system as the inner loop structure. Same simulation requirements as ICHOSE = 9.
16-APR LOCP	Lateral position (y) localizer outer loop command system using the ROLL HOLD command system as the inner loop structure. Same simulation requirements as ICHOSE = 9.

METERS — Logical scalar indicating when units of control law should be changed to meters and radians.

HAC, DAC — Matrices packed by columns into one-dimensional arrays. These matrices are determined in HADAC.

NHAC, NDAC — Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N.

GR — Gravity in ft/sec^2.

N, M, L,
NM, MM — Integer values indicating initial values for dimension of matrices, N = 12, M = 4, L = 4, NM = 2, MM = 2. Changed upon return.

GKPSI Outer loop yaw angle guidance gain for HDG SEL and APR LOCR autopilot modes.

SKY Outer loop lateral position guidance gain for APR LOCR and APR LOCP autopilot modes.

XL, YL Label arrays for printout in DPIFS. Some of the elements in the array are altered upon return.

c. Output Arguments

H, D, FM,
GM, HM, DM Matrices packed by columns into one-dimensional arrays.

NH, ND,
NFM, NGM,
NHM, NDM Two-dimensional integer vector holding the number of rows and columns after the letter N.

XR, UR, YR,
XMR, UMR Logical arrays containing TRUE and FALSE information for subroutine REMAT.

N, M, L,
NM, MM Integer values indicating maximum dimensions for matrices.

XL, YL Label values for plots matching choise of AUTOPILOT mode.

d. Common Blocks

None

e. Error Messages

If the W9 matrix is too small the statement "THE W9 MATRIX IS TOO SMALL" is printed.

if ICHOSE = 1 and ICH(2,9) is FALSE or ICH(2,13) is FALSE then "ERROR ?? ICH BETA BAD" is printed.

If DEBUG is TRUE the following matrices are printed out during the course of program execution.

VXYZ - A 3 x N matrix containing the $\Delta\dot{x}$, $\Delta\dot{y}$, $\Delta\dot{z}$ dynamics extracted from the F matrix.

LAMD - A 1 x N vector containing the row in F that represents the $\Delta\dot{d}$ dynamics.

H(HCON), D(DCON), FM(FMCO), GH(GMCO), HM(HMCO), DM(DMCO)

f. Subroutines Employed by HDCON

ORACLS - MULT, NULL, PRNT, SCALE

PIFLIB - PRNTITL, PRNT

g. Subroutines Employing HDCON

PIFLIB - PIFCGT

h. Concluding Remarks

None

V. INTERPOLATE AERODYNAMIC DATA (INTERP)

1. PURPOSE

Given the aerodynamic coefficients and trim conditions of two flight conditions at different forward velocities, the subroutine INTERP interpolates between the data points to find estimated intermediate aerodynamic coefficients and trim conditions.

2. USAGE

a. Calling Sequence

CALL INTERP (VTP)

b. Input Arguments

VTP Real scalar containing the forward velocity in fps at which the interpolation is to occur

c. Output Arguments

None

d. COMMON Blocks

AERO, AER1

The interpolation overwrites the values in the AERO common block. The AERO common block is used elsewhere in the program scalled by PIFCGT

e. Error Messages

None

f. Subroutine Employed by INTERP

PIFLIB - PRNTITL

g. Subroutines Employing INTERP

PIFCGT

h. Concluding Remarks

None

W. MULTIPLY A TRANSPOSE TIMES B TIMES A (MATBA)

1. PURPOSE

Form Matrix Product A^TBA where B must be symmetric.

2. USAGE

a. Calling Sequence

CALL MATBA (A,NA,B,NB,C,NC,W1,NW1)

b. Input Arguments

A,B Input matrices packed by columns in one-dimensional arrays; not destroyed. B must be symmetrical.

NA,NB Two-dimensional vectors giving number of rows and columns of respective matrices; for example:

NA(1) = number of rows of A

NA(2) = number of columns of B

Not destroyed upon return.

W1 Input workspace vector of length NB(1) x NA(2) or greater

NW1 Vector of length two for workspace

c. Output Arguments

C Product A^TBA packed by columns in one-dimensional array.

NC Two-dimensional vector containing rows and columns in C.

NC(1) = NA(2)

NC(2) = NA(2)

d. COMMON Blocks

None

e. Error Messages

None

f. Subroutines employed by MATBA:

MULT

g. Subroutines employing MATBA:

PIFLIB - CFWM

h. Concluding Remarks

B is not checked to see if it is symmetric.

X. REARRANGE MATRIX (MOVEALL)

1. PURPOSE

Subroutine MOVEALL rearranges the rows (columns) of a matrix with the option of taking a subset of the rows (columns).

2. USAGE

a. Calling Sequence

CALL MOVEALL (A,NA,B,NB,IDEX,IHOW)

b. Input Arguments

A — Matrix to which manipulations are to be performed packed by columns in one dimensional array; not destroyed.

NA — Two-dimensional vector giving number of rows and columns in A.

NA(1) = Number of rows in A

NA(2) = Number of columns in A

Not destroyed.

IHOW — Flag for type of manipulation IHOW = 1 - work with rows, IHOW = 2 - work with columns.

IDEX — Position dependent index vector. IDEX(1) contains the number of the row (column) in A that will become the first row (column) in B and so forth. All unused positions in IDEX must be zero especially when working columns. Therefore if IDEX has its first 5 elements non-zero then B will contain only 5 rows (columns) from A. IDEX thus allows rearranging and deleting rows (columns) at the same time.

c. Output Arguments

B — Contains the manipulated results from A packed by columns in a one-dimensional array.

NB Two-dimensional vector contain dimensions of B

NB(1) = Number of rows in B

NB(2) = Number of columns in B

d. COMMON Blocks

None

e. Error Messages

If NA(1) < NB(1) or NA(2) < NB(2), the message "DIMENSION ERROR IN MOVEALL NA = _____ _____, NB = _____ _____" is printed and the program is returned to the calling point.

f. Subroutines employed by MOVEALL

None

g. Subroutines employing MOVEALL

PIFLIB - REMAT, FGAERO

h. Concluding Remarks

Example: $IDEX = \begin{bmatrix} 3, & 1, & 2, & 0 \end{bmatrix}$

$IHOW = 1$

$$A = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ n & p & q & r \\ t & v & w & x \end{bmatrix}$$

then

$$B = \begin{bmatrix} n & p & q & r \\ a & b & c & d \\ e & f & g & h \end{bmatrix}$$

It is good practice to zero out IDEX before constructing it since non-zero elements detected after a zero element causes problems.

Y. PIF EIGENVALUE PLACEMENT (PIFEIG)

1. PURPOSE

The objective of PIFEIG is to compute a closed-loop model matrix, FM, with stable eigenvalues. Subroutine PIFEIG computes a desirable closed-loop model matrix for the implicit model following design method in the subroutine PIFPLC. Ultimately, PIFPLC computes an initial stabilizing feedback gain for the subroutine DREG.

If ICH(2,11) is TRUE, PIFEIG computes a random feedback gain, GK, using the subroutine RANF. If ICH(2,11) is FALSE, PIFEIG assumes GK contains an initial starting gain. The eigenvalues and eigenvectors of the closed-loop plant, F+G*GK are computed using the subroutine FREEF and satisfy the following relationship

$$T \lambda T^{-1} = F + G * GK \tag{1}$$

λ is a complex diagonal matrix containing the eigenvalues, λ_i. T is a complex matrix containing the eigenvectors. If IE(i) is equal to 1, the i^{th} eigenvalue of λ is adjusted as follows:

eigenvalue $\lambda_i = a_i \pm j\, b_i$

stabilized $a_i = -|a_i|$

After eigenvalue stabilization, the real and imaginary parts of the eigenvector matrix, T=W9(LW8)±jW9(LW10), are used to form the real matrix, FM, as follows,

$$\underbrace{\left[\begin{array}{c|c} W9(LW8) & W9(LW10) \\ \hline -W9(LW10) & W9(LW8) \end{array}\right]}_{ETMA} \underbrace{\left[\begin{array}{c|c} W9(LW1) & W9(LW2) \\ \hline -W9(LW2) & W9(LW1) \end{array}\right]}_{EIMA} \underbrace{\left[\begin{array}{c|c} W9(LW8) & W9(LW10) \\ \hline -W9(LW10) & W9(LW8) \end{array}\right]^{-1}}_{ETMA^{-1}} =$$

$$\left[\begin{array}{c|c} FM & 0 \\ \hline 0 & FM \end{array}\right] \tag{2}$$

The matrix W9(LW1) is a diagonal matrix containing the real part of the stabilized eigenvalues of F+G*GK. The matrix W9(LW2) is a diagonal matrix containing the imaginary part of the stabilized eigenvalues. The partitioned matrix manipulations make it possible to perform complex arithmetic without using complex numbers. The constructed FM matrix has the stable adjusted eigenvalues and the same eigenvectors as F+G*GK

2. USAGE

a. Calling Sequences

CALL PIFEIG (F, NF, FM, NFM, W9, NW9DIM, DEBUG, IFREIG, ER, EI, ERV, IEV, G, NG, ICH, GK, NGK)

b. Input Arguments

F, G, GK — Matrices packed by columns into one-dimensional arrays; F and G are not destroyed upon return. If ICH (2,11) is FALSE, GK contains an initial starting gain and is not destroyed upon return.

IE — Integer vector of dimension at least NF(1). If the eigenvalue adjustment procedure in PIFEIG is to be performed on the i^{th} eigenvalue then IE(i) must be 1. If IE(i) is 0, the i^{th} eigenvalue in FM equals the i^{th} eigenvalue of F+G*GK. If the i^{th} eigenvalue is complex and the i^{th}+1 eigenvalue is the complex conjugate,then IE(i) and IE(i+1) must contain the same integer number (1or 0). The order in IE must apriori agree with the order of the eigenvalues computed by FREEF. The IE feature is useful if a plant eigenvalue is stable but uncontrollable. If ICH(2,11) is TRUE, all elements in IE are automatically set to 1.

ER, EV	Real work vectors of dimension at least NF(1). During computation the adjusted real parts of the eigenvalues are placed in ER and the imaginary parts of the eigenvalues are placed in EV.
IEV	Integer vector of dimension at least NF(1). Each of the locations must contain the integer 1. If ICH(2,11) is TRUE, all IEV elements are automatically set to 1.
ERV	Real work matrix of dimension NF(1)*NF(1). ERV is not used by PIFEIG as long as IEV has all 1's.
W9	Work matrix packed by columns into a one-dimensional array. Dimension must be at least 16*(NF(1)*NF(2)+1)
ICH	Two-dimensional logical matrix dimensioned 10 x 15. Subroutines in PIFCGT are called depending on the TRUE or FALSE status of elements in ICH.
NF NG NGK	Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N.
DEBUG	Logical scalar indicating the following: TRUE: Print out debug information FALSE: Do not print out debug information
IFREIG	The results of the construction of FM can be observed by setting this variable to 0, 1, or 2 2- Compute and print eigenvalues and eigenvectors of FM 1- Compute and print eigenvalues of FM 0- No eigenvalue computations

c. Output Arguments

FM Matrix packed by column into a one-dimensional array. FM contains the desired closed-loop PIF plant used in PIFPLC.

NFM Two-dimensional integer vector holding the number of rows and columns of the matrix FM, NFM(1)=NF(1), NFM(2)=NF(2).

d. COMMON Blocks

None

e. Error Messages

If the W9 matrix is insufficiently large, the message "The W9 matrix is too small" is printed and the program stops. If DEBUG is TRUE, the following matrices are printed out during the course of the computation

GKRA - the random or input GK gain matrix

REAL - the real part of the eigenvalues of F+G*GK

IMAG - the imaginary part of the eigenvalues of F+G*GK

ETMA - the 2*NF(1) x 2*NF(1) matrix containing the partitioned eigenvectors of F+G*GK

ER - the adjusted real part of the eigenvalues of F+G*GK

EI - the adjusted imaginary parts of the eigenvalues of F+G*GK

REAL - the real part of the eigenvalues to be placed in FM

IMAG - the imaginary part of the eigenvalues to be placed in FM.

EIMA - the 2*NF(1) x 2*NF(1) matrix containing the partitioned adjusted eigenvalues of FM.

IDEN - the inverse of ETMA is multiplied by ETMA and the result is printed out as the matrix IDEN.

The accuracy of the inversion can be determined by judging how close IDEN is to an identity matrix.

TLTI - the 2*NF(1) x 2*NF(1) matrix obtained by multiplying ETMA*EIMA*ETMA. If the subroutine GELIM produces an error in inverting ETMA, the message "GELIM IN EIGP IERR = XXXX" is printed

f. Subroutines Employed by PIFEIG

ORACLS - NULL, EQUATE, MULT, ADD, GELIM, UNITY, SCALE, DETFAC, EIGEN

PIFLIB - DIAGPAR, DIAGPUT, CORNER, FREEF, RESPON, SMALL, PRNT, PRNTITL

g. Subroutines Employing PIFEIG

PIFGCT

h. Concluding Remarks

In all applications of PIFEIG by the program PIFCGT, the locations in IE and IEV are all set to 1. It is not necessary to use PIFEIG and PIFPLC if DSTAB in ASYMREG in DREG finds an initial stabilizing gain for the closed-loop plant.

Z. PIF CLOSED-LOOP EIGENVALUE EIGENVECTOR CALCULATIONS (PIFFRE)

1. PURPOSE

PIFFRE takes the matrix log of the closed-loop PIF system matrix, PHICL, then computes the eigenvalues and eigenvectors of the resulting matrix.

2. USAGE

a. Calling Sequences

CALL PIFFRE (DELT, W9, DEBUG, NW9DIM, PHICL, NPHC, IFREEP, ICH, ELOG, MLOG)

b. Input Arguments

PHICL Matrix packed by columns into a one dimensional array; not destroyed upon return.

NPHC Two-dimensional integer vector holding the number of rows and columns of the matrix PHICL.

ELOG Real scalar convergence test for LOGX1. Recommend 1.0E-10.

MLOG Integer scalar indicating the maximum number of iterations used in the LOGX1 subroutine. Recommend 8.

IFREEP Integer scalar used to determine eigenvalue eigenvector calculations

=0 Do not perform calculations

=1 Compute eigenvalues

=2 Compute eigenvalues and eigenvectors

W9 Work matrix packed by columns into a one-dimensional array. Dimension must be at least 3*(NPHC(1)**2+1)

ICH Two-dimensional logical matrix dimensioned 10 x 15. Subroutines in PIFCGT are called depending on the TRUE or FALSE status of elements in ICH.

DEBUG Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug information

DELT Real scalar containing the sampling time

NW9DIM Integer scalar containing the maximum dimension of the W9 matrix.

c. Output Arguments

None

d. COMMON Blocks

None

e. Error Messages

If the W9 matrix is insufficiently large, the message "THE W9 WORK MATRIX IS TOO SMALL" is printed and the program stops.

f. Subroutines Employed by PIFFRE

ORACLS - EQUATE, EXPADE, MULT, NORMS, SCALE, SUBT, UNITY, EIGEN

PIFLIB - LOGX1, FREEF, RESPON

g. Subroutines Employing PIFFRE

PIFCGT

h. Concluding Remarks

None

AA. PIF GAIN COMPUTATIONS (PIFG)

1. PURPOSE

Subroutine PIFG performs three functions:

- o The feedback gain, Riccati equation solution, and feedforward matrix solution are used to construct the matrices in the implementable PIF control law.
- o The body-axis aircraft states used for feedback are replaced with available measurements. Gain matrices in the implementable PIF control law are reconfigured to accommodate the measurements.
- o IF METERS is TRUE, the units in the PIF control law are changed from feet and degrees to meters and radians. Gain matrices in the implementable PIF control law are reconfigured to accommodate the change in units and printed out. The actual values of the PIF control law gains do not change units at the end of PIFG execution if METERS is TRUE. Program execution proceeds as follows:

The feedback gain, GK, is partitioned into three gain sets and they are extracted from GK.

$$GK = \left[\ K1 \ \middle| \ K2 \ \middle| \ K3\ \right] \tag{1}$$

The Riccati equation solution, X, is partitioned into nine individual matrices and three of them (PZZ, PUZT, PXZT) are extracted from X.

$$X = \begin{bmatrix} PXX & PXU & PXZ \\ PXUT & PUU & PUZ \\ PXZT & PUZT & PZZ \end{bmatrix} \tag{2}$$

The matrix A is computed

$$A = PZZ^{-1} * (PUZT * S22 + PXZT * S12) \tag{3}$$

and used to construct S42

$$S42 = K3 * A - K2 * S22 - K1 * S12 \tag{4}$$

The matrix HAC is computed

$$HAC = -K2 * S21 \tag{5}$$

The new measurements, z, are related to the aircraft states by constructing the T matrix,

$$\begin{bmatrix} \underline{z} \\ \underline{u} \end{bmatrix} = \begin{bmatrix} HX & DX \\ 0 & I \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{u} \end{bmatrix} \tag{6}$$

$$T = \begin{bmatrix} HX & DX \\ 0 & I \end{bmatrix} \tag{7}$$

The feedforward matrices are reconfigured to accommodate the new measurements

$$\begin{bmatrix} S11 & S12 \\ S21 & S22 \end{bmatrix}_{new} = \begin{bmatrix} HX & DX \\ 0 & I \end{bmatrix} \begin{bmatrix} S11 & S12 \\ S21 & S22 \end{bmatrix}_{old} \tag{8}$$

The feedback gains K1 and K2 are reconfigured to accommodate the new measurements

$$[K1 \quad K2]_{new} = [K1 \quad K2]_{old} \begin{bmatrix} HX & DX \\ 0 & I \end{bmatrix}^{-1} \tag{9}$$

The PIF gains S31 and S41 are constructed

$$S31 = I + DELT * K2 \tag{10}$$

$$S41 = DELT * K3 \tag{11}$$

The resulting implementable PIF structure has the following form

$$\underline{u}_{k+1} = \underline{u}_k + DELT * \underline{v}_k + S21\ (\underline{u}_{m,k+1} - \underline{u}_{m,k}) \tag{12}$$

$$\underline{e}_k = \underline{x}_k - S11\ \underline{x}_{m,k} \tag{13}$$

$$\begin{aligned} \underline{v}_k = {} & S31\ \underline{v}_{k-1} + HAC(\underline{x}_{m,k} - \underline{x}_{m,k-1}) + S32(\underline{e}_k - \underline{e}_{k-1}) \\ & + S41(H\underline{x}_{k-1} + D\underline{u}_{k-1} - H_m\underline{x}_{m,k-1} - D_m\underline{u}_{m,k}) + S42(\underline{u}_{m,k+1} - \underline{u}_{m,k}) \end{aligned} \tag{14}$$

The gain elements in the implementable PIF structure are modified if the METERS option is TRUE and printed out. The gain matrices at the end of the program execution remain in the units of feet and degrees.

2. USAGE

a. Calling Sequence

CALL PIFG (S11, NS11, S12, NS12, S21, NS21, S22, NS22, GK, NGK, X, NX, S31, NS31, S32, NS32, HAC, NHAC, S41, NS41, S42, NS42, DELT, DEBUG, W9, NW9DIM, HX, DX, NHX, NDX, METERS, ICHOSE)

b. Input Arguments

S11, S12,

S21, S22,

GK, X,

HX, DX — Matrices packed by columns into one-dimensional arrays; HX, DX, S21, and S22 are not destroyed upon return

NS11, NS12,

NS21, NS22,

NGK, NX,

NHX, NDX Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N.

METERS Logical scalar indicating when units of the control law should be changed to meters and radians.

ICHOSE Integer scalar which indicates which command system is being designed.

W9 Work matrix packed by columns into a one-dimensional array. Dimension must be at least 9*(NX(1)*NX(2)+1).

DEBUG Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug inforamtion.

DELT Real scalar containing the sampling time.

NW9DIM Integer scalar containing the maximum dimension of the W9 matrix.

c. Output Arguments

S11, S31,

S32, HAC,

S41, S42,

S21 Matrices packed by columns into one-dimensional arrays constituting the PIF control law feedback and feedforward gains.

NS11, NS31,

NS32, NHAC,

NS41, NS42,

NS21 Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N.

d. COMMON blocks

None

e. Error Messages and Printout

If the W9 matrix is insufficiently large the message "THE W9 WORK MATRIX IS TOO SMALL" is printed and the program stops. If the value in ICHOSE is not a recognized command system then the following is printed "IN PIFG WITH METERS TRUE, ICHOSE = XXX IS NOT A KNOWN COMMAND MODE. CHANGE PIFG TO ACCOMMODATE ICHOSE VALUE." IF DEBUG is TRUE, the following matrices are printed during the course of program execution: K1, K2, K3, PZZ, PZZ^{-1}(PZ-1), PUZT, PXZT, A, T, T^{-1}(T-1)

S11B - The S11 feedforward matrix before the new measurements are accommodated.

S12B - The S12 feedforward matrix before the new measurements are accommodated.

The following are always printed out:

Before changing units

S11, S12, S21, S22, S31, S32, S41, S42, HAC

After changing units if METERS is TRUE

S31M, S21M, HACM, S32M, S11M, S41M, S42M

f. Subroutines Employed by PIFG

ORACLS - MULT, ADD, SUBT, SCALE, UNITY, GELIM, JUXTC, EQUATE, DETFAC

PIFLIB - DIAGPAR, CORNER, PRNT, PRNTITL

g. Subroutines Employing PIFG

PIFLIB - PIFCGT

h. Concluding Remarks

The derivation used to obtain Eqs. 3 to 14 is shown in Chapter III of Ref. 1.

BB. CONSTRUCT THE PIF MODELS (PIFMODL)

1. PURPOSE

The purpose of PIFMODL is to construct the following matrices

$$PHI = \begin{bmatrix} F & G & 0 \\ 0 & 0 & 0 \\ H & D & 0 \end{bmatrix} \tag{1}$$

$$GAMA = \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix} \tag{2}$$

$$QHAT = \begin{bmatrix} A & AM & 0 \\ AM^T & R & 0 \\ 0 & 0 & QZ \end{bmatrix} \tag{3}$$

$$RHAT = RD \tag{4}$$

$$HX = I \tag{5}$$

$$DX = 0 \tag{6}$$

HX and DX are NF(1) and NF(1) x NG(2) aircraft observation matrices. The order of the states is Δu Δw Δq $\Delta\theta$ Δz Δx Δv Δr Δp $\Delta\phi$ $\Delta\psi$ Δy. If ICH(2,10) is TRUE, Δu observation in HX and DX is replaced with Δa_x observation. If ICH(2,9) is TRUE, Δv observation in HX and DX is replaced with Δa_y observation. If ICH(2,10) is TRUE, Δw observation in HX and DX is replaced with Δa_z observation. Variables in XL are changed depending on whether or not METERS, ICH(2,8), ICH(2,9) and ICH(2,10) are TRUE.

2. USAGE

a. Calling Sequences

CALL PIFMODL(F, NF, G, NG, H, NH, D, ND, PHI, NPH, GAMA, NGA, Q, NQ, R, NR, AM, NAM, QZ, NQZ, QHAT, NQH, RD, NRD, RHAT, NRH, ZERO, NZERO, HX, NHX, DX, NDX, HAC, NHAC, DAC, NDAC, W9, NW9DIM, DEBUG, METERS, ICH, XL, UL, YL)

b. Input Arguments

F, G,

H, D,

Q, AM,

R, QZ,

RD — Matrices packed by columns into one-dimensional arrays; not destroyed upon return.

HAC, DAC — Matrices packed by columns into one-dimensional arrays; not destroyed upon return. These are the observation matrices for the body mounted accelerometers computed in HADAC.

XL, UL, YL — Two dimensional arrays containing plotting labels for the states, controls, and commanded outputs, respectively.

NF, NG, NH,

ND, NQ, NR,

NAM, NQZ,

NZERO, NHAC,

NDAC — Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N.

W9 — Work matrix packed by columns into a one-dimensional array. Dimension must be at least 2 x {(NF(1) + NG(2) + NH(1))**2+1}

ZERO — Work matrix packed by columns into a one-dimensional array.

All elements must be 0.0. Dimensions must be at least NPH(1) * NPH(2)+1

ICH Two-dimensional logical matrix dimensioned 10 x 15. Subroutines in PIFCGT are called depending on the TRUE or FALSE status of elements in ICH.

DEBUG Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug information

NW9DIM Integer scalar containing the maximum dimension of the W9 matrix.

c. Output Arguments

QHAT, RHAT,

HX, DX,

PHI, GAMA Matrices packed by columns into one dimensional arrays.

NQH, NRH,

NPH, NGA Two dimensional integer vectors holding the number of rows and columns of the matrix shown after the letter N.

XL Two dimensional array containing labels to be used in plotting; some are changed upon return depending on logical parameters METERS and ICH.

d. COMMON Blocks

None

e. Error Messages

If the W9 matrix is insufficiently large the message "THE W9 WORK MATRIX IS TOO SMALL" is printed and the program stops. If DEBUG is true then the following is printed out during program execution:

PHI, GAMA, QHAT, RHAT, HX, DX

f. Subroutines Employed by PIFMODL

ORACLS - JUXTC, UNITY, JUXTR, TRANP, EQUATE

PIFLIB - DIAGPAR, CORNER, PRNT, PRNTITL

g. Subroutines employing PIFMODL

PIFCGT

h. Concluding Remarks

The perturbation relationship between the body mounted accelerometer outputs, Δa_x Δa_y and Δa_z, and the aircraft body-axis states is derived in Ref. 1, Appendix B.

CC. PIF EIGENVALUE PLACEMENT (PIFPLC)

1. PURPOSE

In Ref. 6 a technique is given for computing a feedback gain which places eigenvalues and eigenvectors of a plant to those dictated by an ideal model. The purpose of PIFPLC is to compute a stabilizing feedback gain for the PIF control law using the eigenvalue eigenvector placement procedure in Ref. 6. The ideal model (PHICL) is computed in PIFEIG. The subroutine is only applicable to PIF model matrices. The subroutine first extracts the H matrix from the discrete PIF model matrix PHI. The discrete feedforward matrix equation

$$\begin{bmatrix} PHI & GAMA \\ H & 0 \end{bmatrix} \begin{bmatrix} S11 \\ S21 \end{bmatrix} = \begin{bmatrix} S11 & PHICL \\ & H \end{bmatrix} \tag{1}$$

is solved for S11 and S21 using CGTPIF. The stabilizing feedback gain is computed as

$$GK = S21 \quad S11^{-1} \tag{2}$$

2. USAGE

a. Calling Sequence

CALL PIFPLC (PHI, NPH, NH, GAMA, NGA, W9, NW9DIM, X, PHICL, NPHC, ACL, GK, NGK, DEBUG, ELOG, MLOG, ICH, DELT, NS11, NS12, NS21, NS31 NS32, NS41, NS42, ICGT, TOL, ITER, ECGT)

b. Input Arguments

PHI, GAMA,

PHICL Matrices packed by columns into one-dimensional arrays;

not destroyed upon return. PHI contains the discrete PIF system matrix, GAMA contains the discrete PIF control effect matrix, and PHICL contains the desired closed-loop plant matrix (FM in PIFEIG)

NS11, NS12, NS21, NS22, NS31, NS32, NS41, NS42	Two dimensional work integer vectors.
ELOG	Real scalar convergence test for LOGX1 subroutine.
X,ACL	Work matrices of dimension at least NPH(1) * NPH (2) + 1
W9	Work matrix packed by columns into a one-dimensional array. Dimension must be at least 9*(NPH(1)**2+1) + 4 *((NPH(1) + NGA(2) + 2)**2+1) + 2*((NPH(1) + NGH(2) + 1)**2+1)
ICH	Two-dimensional logical matrix dimensioned 10 x 15. Subroutines in PIFCGT are called depending on the TRUE or FALSE status of elements in ICH.
NH, NPH, NGA, NPHC	Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N.
TOL	Real scalar indicating convergence criteria for iterative refinement in SOLVER. Recommend 1.0E-10
ITER	Integer scalar indicating maximum number of interative refinements in SOLVER. Recommend 10.
ECGT	Real scalar indicating convergence criteria in SOLVER for inverting a matrix. Recommend 1.0E-10
ICGT	Five dimensional integer option vector defined in CGTPIF.

ICGT(1) is set to 2, ICGT(4) is set to 0 and ICGT(5) is set to 0 automatically by the subroutine before calling PIFCGT. The orginial values of ICGT are restored at the end of subroutine execution.

DEBUG Logical scalar indicating the following

TRUE: Print out debug information

FALSE: Do not print out debug information

DELT Real scalar containing the sampling time.

NW9DIM Integer scalar containing the maximum dimension of the W9 matrix.

c. Output Arguments

GK Matrix packed by columns into a one-dimensional array containing the stabilizing feedback gain.

NGK Two dimensional integer vector holding the number of rows and columns of the GK matrix.

d. COMMON Blocks

None

e. Error Messages

If the W9 matrix is insufficiently large the message "THE W9 MATRIX IS TOO SMALL" is printed and the program stops. If the subroutine GELIM produces an error in inverting S11, the message "GELIM ERROR IN PIFPLC IERR = XX" is printed. If DEBUG is TRUE then the following is printed out during the course of program execuation:

HDO - The H matrix extracted from the PIF model matrix PHI.

Matrices from the execution of CGTPIF (see CGTPIF document).

S11G - The S11 feedforward matrix solution of the feedforward matrix equation.

S21G - The S21 feedforward matrix solution of the feedforward matrix equation.

IØØG - S11 is multiplied times $S11^{-1}$ and the results, which should be an identity matrix, is printed out.

The equvalent continuous-time closed-loop eigenvalues and eigenvectors of PHI + GAMA * GK.

f. Subroutines Employed by PIFPLC

ORACLS - NULL, EQUATE, UNITY, GELIM, MULT, ADD, DETFAC, SCALE, SUBT, NORMS, BCKMLT, HSHLDR, SCHUR, SYSSLV, EXPADE, EIGEN

PIFLIB - PRNT, PRNTITL, EXTR, FREEF, RESPON, LOGX1, CGTPIF, AXBMXC, SHRSOL

g. Subroutines Employing PIFPLC

PIFLIB - PIFCGT

h. Concluding Remarks

It is not necessary to use PIFEIG and PIFPLC if DSTAB in ASYMREG in DREG finds an initial stabilizing gain.

DD. OUTPUT NUMERICAL DATA (PRNT)

1. PURPOSE

Subroutine PRNT prints a single matrix with a descriptive heading on the same page. The descriptive heading is printed before each matrix and is of the form "NAM MATRIX NC(1) ROWS NA(2) COLUMNS." The matrix is printed by rows for seven columns at a time using the format, (I4,1P7E16.7), where the first number is an integer depicting the row number. A heading of integers corresponding to the column number is printed above the matrix.

2. USAGE

a. Calling Sequence

CALL PRNT (A,NA,NAM,IOP)

b. Input Arguments

A   Matrix packed by columns in a one-dimension array

NA   Two-dimensional vector giving the number of rows and columns of the matrix A:

NA(1) = Number of rows of A

NA(2) = Number of columns of A

NAM   Hollerith characters giving matrix name. Generally NAM should contain 4 Hollerith characters and be written in the argument list as 4HXXXX.

IOP   Scalar print control parameter

1. print heading and matrix on same page
2. print heading and matrix on next page

c. Output Arguments

None

d. COMMON Blocks

None

e. Error Messages

If NA(1) < 1 or NA(2) < 1 the message "PROBLEM WITH NUMBER OF ROWS OR NUMBER OF COLUMNS NA(1) = _____ _____, NA(2) = _____ _____" is printed, and the program is returned to the calling point.

f. Subroutine Employed by PRNT

None

g. PIFLIB subroutines Employing PRNT

PIFCGT, CGTPIF, HDCON, CFWM, PIFMODL, PIFPLC, REMAT, DISCMOD, QRMP1, QRMHAT, DREG, PIFG, DPIFS, FGAERO, HADAC, PIFEIG

h. Concluding Remarks

None

EE. PRINT TITLE (PRNTITL)

1. PURPOSE

Subroutine PRNTITL is used to print the name of a subroutine at the beginning and end of each pass through the subroutine.

2. USAGE

a. Calling Sequence

CALL PRNTITL (NAM1, NAM2)

b. Input Arguments

NAM1 Hollerith characters for title to be printed out. The first ten Hollerith characters in the title are placed in NAM1 in the form 10HXXXXXXXXXX.

NAM2 Hollerith characters for title to be printed out. The second ten Hollerith characters in the title are placed in NAM2 in the form 10HXXXXXXXXXX.

c. Output Arguments

None

d. COMMON Blocks

None

e. Error Messages

None

f. Subroutines Employed by PRNTITL

None

g. PIFLIB subroutines Employing PRNTITL

DIMSS, HDCON, CFWM, PIFMODL, PIFPLC, REMAT, DISCMOD, QRMP1, QRMHAT, DREG, PIFG, DPIFS, FGAERO, HADAC, PIFEIG, RUNINFO, INTERP, PIFFRE

h. Concluding Remarks

None

FF. OUTPUT LABELED NUMERICAL DATA (PRNT2)

1. PURPOSE

Subroutine PRNT2 prints out one matrix in columns of seven with labels or numbers, or it prints out two matrices in columns of seven with labels for comparison. PRNT2 also has the option of printing a third one dimensional array as a counter instead of row numbers.

2. USAGE

a. Calling Sequence

```
CALL PRNT2  (A, NA, NAM, B, NB, NAMB, IOP, C, LABEL, LABEL1, LABEL2, M)
```

b. Input Arguments

A — Matrix packed by columns into a one-dimensional array. A is the first matrix to be printed.

B. — Matrix packed by columns into a one-dimensional array. B is the second matrix to be printed.

NA — Two-dimensional vector giving the number of rows and columns of matrix A.

NB — Two-dimensional vector giving the number of rows and columns of matrix B.

NA(1) - number of rows of A

NA(2) - number of columns of A

NB(1) - number of rows of B = NA(1)

NB(2) - number of rows of B = NA(2)

NAM — Hollerith characters giving matrix A's name. Generally NAM should contain 4 Hollerith characters and be written in the argument list as 4HXXXX.

NAMB This is the name of the matrix B and follows the same rules as NAM.

IOP Scalar print control parameter.

IOP = 0	PRINT ONE MATRIX WITH LABELS - SAME PAGE
IOP = 1	PRINT ONE MATRIX WITH LABELS - NEW PAGE
IOP = 2	PRINT ONE MATRIX WITH NUMBERS - SAME PAGE
IOP = 3	PRINT ONE MATRIX WITH NUMBERS - NEW PAGE
IOP = 4	PRINT TWO MATRICES WITH LABELS - SAME PAGE
IOP = 5	PRINT TWO MATRICES WITH LABELS - NEW PAGE

C One-dimensional counter array; dimension = NA(1)

LABEL LABELS for Array A, dimensioned (M,2)

LABEL1 LABELS for Array B, dimensioned (M,2)

LABEL2 LABELS for Array C, dimensioned (1,2)

M First Dimension of LABEL array = NA(2)

c. Output Arguments

None

d. COMMON Blocks

None

e. Error Messages

If NA(1) < 1 or NA(2) < 1 the message "PROBLEM WITH THE NUMBER OF ROWS OR NUMBER OF COLUMNS NA(1) = _____, NA(2) = _____" is printed, and the program is returned to the calling point.

f. Subroutines Employed by PRNT2

None

g. Subroutines Employing PRNT2

PIFLIB - DPIFS

h. Concluding Remarks

For printing one matrix - B, NB, NAMB and LABEL1 are dummy arguments, but they must be in the call to PRNT2. For printing without labels - C, LABEL, LABEL1, and LABEL2 are dummy arguments, but they must be in the call to PRNT2.

GG. SAMPLED-DATA REGULATOR USING ALPHA AND BETA (QRMHAT)

1. PURPOSE

The purpose of QRMHAT is to compute the sampled-data regulator corresponding to a linear time-invariant continuous system and its associated linear quadratic cost function

$$\dot{\underline{x}} = F\underline{x} + G\underline{u} \tag{1}$$

$$J = \int_0^\infty \underline{x}^T Q\underline{x} + \underline{u}^T R\underline{u}\ dt \tag{2}$$

The control, $\underline{u}$, is assumed to be constant over the sampling interval Δt. The discrete plant and cost function are given by

$$\underline{x}_{k+1} = \mathrm{PHI}\ \underline{x}_k + \mathrm{GAMA}\ \underline{u}_k \tag{3}$$

$$J = \sum_{k=0}^{\infty} (\underline{x}_k^T\ \mathrm{QHAT}\ \underline{x}_k + 2\underline{x}_k^T\ \mathrm{AMHAT}\ \underline{u}_k + \underline{u}_k^T\ \mathrm{RHAT}\ \underline{u}_k) \tag{4}$$

The matrices in the sampled-data regulator problem are computed using the following equations

$$\mathrm{PHI} = \sum_{k=0}^{\infty} \frac{(F\Delta t)^k}{k!} = e^{F\Delta t} \tag{5}$$

$$\mathrm{GAMA} = \int_0^{\Delta t} \sum_{k=0}^{\infty} \frac{(F\tau)^k}{k!}\ d\tau = \int_0^{\Delta t} e^{F\tau}\ d\tau \tag{6}$$

$$\mathrm{QHAT} = \int_0^{\Delta t} e^{F^T\tau}\ Qe^{F\tau}\ d\tau \tag{7}$$

$$\mathrm{AMHAT} = \left[\int_0^{\Delta t} e^{F^T\tau}\ Q(\int_0^{\tau} e^{Fs}\ ds)d\tau\right]G \tag{8}$$

$$\mathrm{RHAT} = \Delta t\ R + G^T \left[\int_0^{\Delta t}(\int_0^{\tau} e^{A^Ts}\ ds)Q(\int_0^{\tau} e^{As}\ ds)d\tau\right]G \tag{9}$$

QHAT is computed using the ORACLS subroutine SAMPL. PHI and GAMA are computed using the ORACLS subroutine EXPINT. AMHAT and RHAT are computed using the subroutine ALPHA. If ICH(2,7) is TRUE, a matrix similar to PHI is computed using the matrix PHICL and the ORACLS subroutine EXPSER. The continuous time model for "F" used in computing PHICL is determined in the subroutine PIFEIG as the matrix FM. The last part of QRMHAT is specially adapted to PIF controller computation. For PIF, F and G in the calling program have the following internal form

$$F = \begin{bmatrix} A & B & 0 \\ 0 & 0 & 0 \\ H & D & 0 \end{bmatrix} \tag{10}$$

$$G = \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix} \tag{11}$$

After PHI and GAMA are computed using EXPINT, they are altered from

$$PHI = \begin{bmatrix} PHI_A & GAMA_B & 0 \\ 0 & I & 0 \\ X_1 & X_2 & I \end{bmatrix} \tag{12}$$

$$GAMA = \begin{bmatrix} X_3 \\ \Delta t I \\ X_4 \end{bmatrix} \tag{13}$$

to

$$PHI = \begin{bmatrix} PHI_A & GAMA_B & 0 \\ 0 & I & 0 \\ \Delta t H & \Delta t D & I \end{bmatrix} \tag{14}$$

$$GAMA = \begin{bmatrix} 0 \\ \Delta t I \\ 0 \end{bmatrix} \tag{15}$$

2. USAGE

a. Calling Sequence

CALL QRMHAT (H, NH, QHAT, NQH, D, ND, RHAT, NRH, AMHAT, NAM, PHI, NPH, GAMA, NGA, DELT, ICH, DEBUG, W9, NW9DIM, ZERO, NZERO, PHICL, NPHC)

b. Input Arguments

H, QHAT,

D, RHAT,

PHI, GAMA,

PHICL — Matrices packed by columns in one-dimensional arrays. H and D are not destroyed upon return. The rest are destroyed upon return. Using the notation in the PURPOSE section, QHAT contains Q, RHAT contains R, PHI contains F, GAMA contains G, and PHICL contains the desired continuous-time closed-loop system (FM) constructed in PIFEIG.

W9 — Work matrix packed by columns into a one-dimensional array. Dimension must be at least 14* (NPH(1) * NPH(2) + 1)

ZERO — Work matrix packed by columns into a one-dimensional array. All elements must be zero. Dimension must be at least NPH(1) * NPH(2) + 1

ICH — Two-dimensional logical matrix dimensioned 10 x 15. Subroutines in PIFCGT are called depending on the TRUE or FALSE status of elements in ICH.

NH, NQH,

ND, NRH,

NAM, NPH,

NGA, NZERO,

NPHC — Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N. NZERO is the only vector destroyed upon return.

DEBUG Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug information

DELT Real scalar containing the sampling time.

NW9DIM Integer scalar containing the maximum dimension of the W9 matrix.

c. Output Arguments

QHAT, RHAT,

AMHAT, PHI,

GAMA, PHICL Matrices packed by columns into one dimensional arrays containing the discrete plant representation shown in the PURPOSE section.

d. COMMON Block

None

e. Error Messages

If the W9 matrix is insufficiently large the message "THE W9MATRIX IS TOO SMALL" is printed and the program stops. If ICH(4,2) and ICH(4,3) are both true the message "BOTH ICH(4,2) AND ICH(4,3) ARE TRUE CONFLICT IN CALCULATIONS OF GAMA, PHI, QHAT, RHAT, AND AMHAT" is printed and the program stops. If DEBUG is true the print option in ORACLS subroutines SAMPL and EXPINT are activated and the following matrices at the end of subroutine execution are printed: PHI, GAMA, QHAT, RHAT, AMHAT, (AMHA), PHICL (PHCQ).

f. Subroutines Employed by QRMHAT

ORACLS - EQUATE, SAMPL, EXPINT, MULT, EXPSER, TRANP, SCALE, ADD, UNITY,
JUXTC, JUXTR

PIFLIB - PRNTITL, PRNT, CORNER, ALPHA, BETA

g. Subroutines Employing QRMHAT

PIFCGT

h. Concluding Remarks

ORACLS software has a problem when computing the sampled data regulator representation when NGA(2) = 1. QRMHAT serves as backup software to QRMP1 when the latter fails.

HH. SAMPLED-DATA REGULATOR USING ORACLS (QRMP1)

1. PURPOSE

The purpose of QRMP1 is to compute the sampled-data regulator corresponding to a linear time-invariant continuous system and its associated linear quadratic cost function.

$$\dot{\underline{x}} = F\underline{x} + G\underline{u} \tag{1}$$

$$J = \int_0^{\infty} \underline{x}^T Q\underline{x} + \underline{u}^T R\underline{u}\, dt \tag{2}$$

The control, $\underline{u}$, is assumed to be constant over the sampling interval, Δt. The discrete plant and cost function are given by

$$x_{k+1} = \text{PHI}\ \underline{x}_k + \text{GAMA}\ \underline{u}_k \tag{3}$$

$$J = \sum_{k=0}^{\infty} (\underline{x}_k^T\ \text{QHAT}\ \underline{x}_k + 2\underline{x}_k^T\ \text{AMHAT}\ \underline{u}_k + \underline{u}_k^T\ \text{RHAT}\ \underline{u}_k) \tag{4}$$

The matrices in the sampled-data regulator problem are computed using the following equations

$$\text{PHI} = \sum_{k=0}^{\infty} \frac{(F\Delta t)^k}{k!} = e^{F\Delta t} \tag{5}$$

$$\text{GAMA} = \int_0^{\Delta t} \sum_{k=0}^{\infty} \frac{(F\tau)^k}{k!}\, d\tau = \int_0^{\Delta t} e^{F\tau}\, d\tau \tag{6}$$

$$\text{QHAT} = \int_0^{\Delta t} e^{F^T\tau} Q e^{F\tau}\, d\tau \tag{7}$$

$$\text{AMHAT} = \left[\int_0^{\Delta t} e^{F^T\tau} Q(\int_0^{\tau} e^{Fs}\, ds) d\tau\right] G \tag{8}$$

$$\text{RHAT} = \Delta t\ R + G^T [\int_0^{\Delta t} \left[(\int_0^{\tau} e^{A^Ts}\, ds) Q (\int_0^{\tau} e^{As}\, ds) d\tau\right] G \tag{9}$$

QHAT, RHAT AND AMHAT are computed using the ORACLS subroutine SAMPL. PHI and GAMA are computed using the ORACLS subroutine EXPINT. If ICH(2,7) is TRUE a matrix similiar to PHI is computed using the matrix PHICL and the ORACLS subroutine EXPSER. The continuous time model for "F" used in computing PHICL is determined in the subroutine PIFEIG as the matrix FM. The last part of QRMP1 is specially adapted to PIF controller computation. For PIF, F and G in the calling program have the following form:

$$F = \begin{bmatrix} A & B & 0 \\ 0 & 0 & 0 \\ H & D & 0 \end{bmatrix} \tag{10}$$

$$G = \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix} \tag{11}$$

After PHI and GAMA are computed they are altered from

$$PHI = \begin{bmatrix} PHI_A & GAMA_B & 0 \\ 0 & I & 0 \\ X_1 & X_2 & I \end{bmatrix} \tag{12}$$

$$GAMA = \begin{bmatrix} X_3 \\ \Delta t I \\ X_4 \end{bmatrix} \tag{13}$$

to

$$PHI = \begin{bmatrix} PHI_A & GAMA_B & 0 \\ 0 & I & 0 \\ \Delta tH & \Delta tD & I \end{bmatrix} \tag{14}$$

$$GAMA = \begin{bmatrix} 0 \\ \Delta t I \\ 0 \end{bmatrix} \tag{15}$$

The alteration implies the particular digital implementation used in the PIF control law as described in Ref. 1, Chapter III.

2. USAGE

a. Calling Sequence

CALL QRMP1 (H, NH, QHAT, NQH, D, ND, RHAT, NRH, AMHAT, NAM, PHI, NPH, GAMA, NGA, DELT, ICH, DEBUG, W9, NW9DIM, ZERO, NZERO, PHICL, NPHC)

b. Input Arguments

H, QHAT,

D, RHAT,

PHI, GAMA,

PHICL — Matrices packed by columns in one-dimensional arrays. H and D are not destroyed upon return. The rest are destroyed upon return. Using the notation in the PURPOSE section, QHAT contains Q, RHAT contains R, PHI contains F, GAMA contains G and PHICL contains the desired continuous-time closed-loop system (FM) constructed in PIFEIG.

W9 — Work matrix packed by columns into a one-dimensional array. Dimension must be at least 14 * (NPH(1) * NPH(2) + 1)

ZERO — Work matrix packed by columns into a one-dimensional array. All elements must be zero. Dimension must be at least NPH(1) * NPH(2) + 1

ICH — Two-dimensional logical matrix dimensioned 10 x 15. Subroutines in PIFCGT are called depending on the TRUE or FALSE status of elements in ICH.

NH, NQH, ND,

NRH, NAM,

NPH, NGA,

NZERO, NPHC, Two-dimensional integer vector holding the number of rows and columns of the matrix shown after the letter N. NZERO is the only vector destroyed upon return.

DEBUG Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug information

DELT Real scalar containing the sampling time.

NW9DIM Integer scalar containing the maximum dimension of the W9 matrix.

c. Output Arguments

QHAT, RHAT,

AMHAT, PHI,

GAMA, PHICL Matrices packed by columns into one dimensional arrays containing the discrete plant representation shown in the PURPOSE section.

d. COMMON Blocks

None

e. Error Messages

If the W9 matrix is insufficiently large the message "THE W9 MATRIX IS TOO SMALL" is printed and the program stops. If DEBUG is true the following matrices at the end of subroutine execution are printed: PHI, GAMA, QHAT, RHAT, AMHAT (AMHA), PHICL (PHCQ)

f. Subroutines Employed by QRMP1

ORACLS - EQUATE, SAMPL, EXPINT, MULT, EXPSER, TRANP, SCALE, ADD, UNITY, JUXTC, JUXTR

PIFLIB - PRNTITL, PRNT, CORNER

g. Subroutines Employing QRMP1

PIFLIB - PIFCGT

h. Concluding Remarks

ORACLS subroutine SAMPL has a problem when computing the sampled data regulator representation when NGA(2) = 1. QRMHAT serves as back up software to QRMP1 when the later fails.

II. REDUCE MATRICES (REMAT)

1. PURPOSE

The purpose of REMAT is to eliminate rows and columns of matrices, and reduce the matrix dimensions. The information used by REMAT to eliminate rows and columns is given in the logical vectors XR, UR, YR, XMR, and UMR. The values in the logical vectors are specified in the subroutine HDCON. If a logical variable entry is TRUE, the corresponding state or control or command is to be retained in the PIF design problem. If a logical variable entry is FALSE, the corresponding state or control or command is to be eliminated. The correspondance between logical vector elements and variables is as follows:

Δu	Δw	Δq	$\Delta\theta$	Δz	Δx	Δv
XR(1)	XR(2)	XR(3)	XR(4)	XR(5)	XR(6)	XR(7)

Δr	Δp	$\Delta\phi$	$\Delta\psi$	Δy
XR(8)	XR(9)	XR(10)	XR(11)	XR(12)

$\Delta\delta_T$	$\Delta\delta_e$	$\Delta\delta_a$	$\Delta\delta_r$
UR(1)	UR(2)	UR(3)	UR(4)

YLON1	YLON2	YLAT1	YLAT2
YR(1)	YR(2)	YR(3)	YR(4)

XM_1	XM_2	XM_3		XM_{NM}
XMR(1)	XMR(2)	XMR(3)	. . .	XMR(NM)

UM_1	UM_2	UM_3		UM_{MM}
UMR(1)	UMR(2)	UMR(3)	. . .	UMR(MM)

XR is used to eliminate rows in the following matrices: F, G, Q, AM, PHI, QHAT, GAMA, XL, HX, HAB

XR is used to eliminate columns in the following matrices: F, Q, H, PHI, QHAT, HX, DX

UR is used to eliminate rows in the following matrices: UL, R, RD, RHAT, PHI, GAMA, QHAT

UR is used to eliminate columns in the following matrices: G, D, R, AM, RD, RHAT, GAMA, PHI, QHAT

UR is used to eliminate rows in the following matrices: YL, H, D, QZ, HM, DM, PHI, GAMA, QHAT

YR is used to eliminate columns in the following matrices: QZ, PHI, QHAT

XMR is used to eliminate rows in the following matrices: FM, GM

XMR is used to eliminate columns in the following matrices: FM, HM

UMR is used to eliminate columns in the following matrices: GM, DM

2. USAGE

a. Calling Sequence

CALL REMAT (N, M, L, NM, MM, DEBUG)

b. Input Arguments

N, M, L,

NM, MM — Real scalars indicating matrix dimensions before reduction; changed upon return

DEBUG — Logical scalar indicating the following:

TRUE: Print out debug information

FALSE: Do not print out debug information

c. Output Arguments

N, M, L,

NM, MM   Real scalars indicating matrix dimensions after reduction.

d. COMMON Blocks

DISCSV, WORKS9, WGHTS, MODF, DISCM, PLA, DIMN1, DIMN2, DIMN3, DIMN4, EIGEN, LABEL

e. Error Messages and Printout

If DEBUG is TRUE then following sequence of matrices are printed out after they have been reduced: F, HX, HAB, Q, G, DX, R, AM, RD, H, D, QZ, FM, HM, GM, DM, PHI, QHAT, GAMA, RHAT

f. Subroutines Employed by REMAT

ORACLS - EQUATE

PIFLIB - PRNT, PRNTITL, MOVEALL

g. Subroutines Employing REMAT

PIFCGT

h. Concluding Remarks

None

JJ. STEP RESPONSE EIGENVALUE INFORMATION (RESPON)

1. PURPOSE

RESPON in association with FREEF and FREEO computes step response information for a given eigenvalue.

2. USAGE

a. Calling Sequence

CALL RESPON (RR, RI, FREQNA, DAMRAT, THALF, PERIOD, TAU, T90, T98, T100, OVERSH)

b. Input Arguments

RR	Real part of an eigenvalue
RI	Imaginary part of an eigenvalue

c. Output Arguments

FREQNA	Natural frequency
DAMRAT	Damping ratio
THALF	Time to reach 50% of a step command
PERIOD	Time for a complex eigenvalue to complete one oscillation
TAU	Time constant
T90	Time to reach 90% of a step command
T98	Time to reach 98% of a step command
T100	Time for the step response of a complex eigenvalue to cross the step command the first time.
OVERSH	The overshoot of a stable complex eigenvalue in response to a step input.

d. COMMON Blocks

None

e. Error Messages

None

f. Subroutines Employed by RESPON

None

g. Subroutines Employing RESPON

PIFLIB - FREEF, FREEO

h. Concluding Remarks

None

KK. PRINT OUT INFORMATION (RUNINFO)

1. PURPOSE

Subroutine RUNINFO prints out the values of all variables used in the PIF design. RUNINFO contains a listing of all available subroutine calls and a listing of all variables used by the PIF program along with their definitions.

2. USAGE

a. Calling Sequence

CALL RUNINFO (NUM)

b. Input Arguments

NUM Directs RUNINFO to print out one of two kinds of information

=1 Print out subroutine call information

=2 Print out variable input information

c. Output Arguments

None

d. COMMON Blocks

WORKS9, RNINFO, RNINFO1, RNINFO2, QWHT, EIGEN, LABLE, UDIR, GLAD, DEBG, AERO, EIGS, HDXA

e. Error Messages

None

f. Subroutines Employed by RUNINFO

None

g. Subroutines Employing RUNINFO

PIFLIB - PIFCGT

h. Concluding Remarks

None

LL. CREATE A (3XN) MATRIX FROM 3 DIFFERENT 1XN MATRICES (R3MAT)

1. PURPOSE

Three row vectors A, B and C are scaled by SC1, SC2, and SC3, respectively. The three row vectors are grouped into the matrix D,

$$D = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \tag{1}$$

The matrix D is scaled by the scalar S.

2. USAGE

a. Calling Sequence

CALL R3MAT (A, B, C, D, ND, NW1, SC1, SC2, SC3, S, W9)

b. Input Arguments

A, B, C — One dimensional vectors

NW1 — Two dimensional integer vector containing the number of rows and columns of A, B and C:

NW1(1) = 1

NW1(2) = N

SC1, SC2,
SC3 — Scale factors for A, B and C, respectively

S — Scale factor for the matrix D after creation

W9 — Work space

c. Output Arguments

D — A 3 x N matrix packed by columns into a one-dimensional array with rows of A, B and C, respectively.

ND — Number of rows and columns of D

ND(1) = Number of rows of D

ND(2) = Number of columns of D

d. COMMON Blocks

None

e. Error Messages

None

f. Subroutines Employed by R3MAT

ORACLS - JUXTR, SCALE

g. Subroutines Employing R3MAT

PIFLIB - FGAERO

h. Concluding Remarks

None

MM. SCHUR FORM SOLUTION OF A * X * B - X = C (SHRSOL)

1. PURPOSE

Subroutine SHRSOL solves the algebraic matrix equation

$$A*X*B - X = C \tag{1}$$

where it is assumed that the matrix A is in lower real SCHUR form and the matrix B is in upper real SCHUR form. Subroutine SYSSLV is used in SHRSOL to solve a system of order 2 or 4 by Crout elimination. Common block SLVBLK is used to communicate with SYSSLV.

2. USAGE

a. Calling Sequence

CALL SHRSOL (A,B,C,N,M,NA,NB,NC)

b. Input Arguments

A,B,C	Matrices stored in two-dimensional array. First dimensions are NA, NB and NC, respectively. Second dimensions are at least N, M, and N. Matrix C is destroyed upon return.
N	Order of the matrix A.
M	Order of the matrix B.
NA,NB, NC	Maximum first dimensions of the matrices A, B, and C, respectively, as given in the DIMENSION statement of the calling program.

c. Output Arguments

C Two-dimensional array containing the solution, X.

d. COMMON Blocks

SLVBLK

e. Error Messages

If the subroutine fails to find a solution the message "ALGORITHM FAILED TO PRODUCE SOLUTION" is printed and the program terminates execution.

f. Subroutines Employed by SHRSOL

ORACLS - SYSSLV

g. PIFLIB subroutines Employing SHRSOL

AXBMXC

h. Concluding Remarks

It is recommended that the program SOLVER which calls AXBMXC which subsequently calls SHRSOL be used to solve Eq. 1. This subroutine is a generalization of algorithms reported in Ref. 3.

NN. ROUND OFF SMALL NUMBERS (SMALL)

1. PURPOSE

To zero out elements in a matrix with very small values.

2. USAGE

a. Calling Sequence

CALL SMALL (A,NA)

b. Input Arguments

A Input matrix packed by columns in a one dimensional array.

NA Two-dimensional vector contains number of rows and columns in A

NA(1) = number of rows in A

NA(2) = number of columns in A

c. Output Arguments

A If $|A(i,j)| < 1 \times 10^{-8}$ then $A(i,j) = 0$. No action otherwise.

NA No change

d. Common Blocks

None

e. Error Messages

None

f. Subroutines employed by SMALL

None

g. Subroutines employing SMALL

PIFLIB - PIFEIG

h. Concluding Remarks

Helps in interpreting output if very small elements are zero.

OO. ITERATIVE REFINEMENT SOLUTION OF A * X * B - X = C (SOLVER)

1. PURPOSE

Subroutine SOLVER solves the algebraic matrix equation

$$A*X*B - X = C \tag{1}$$

for the matrix X using the program AXBMXC. SOLVER is primarily used to determine a highly accurate solution to Eq. 1 using iterative refinement. After AXBMXC computes the first solution to (1), the following matrices are computed using double precision,

$$ERROR = C + X - A*X*B \tag{2}$$

$$A*X1*B - X1 = ERROR \tag{3}$$

If the Euclidean norm of X1 is less than some tolerance (TOL) the iterative refinement procedure is terminated and X is the solution. If the norm is not less then some tolerance, X is updated,

$$X = X + X1 \tag{4}$$

substituted back into Eq. 2 and the iterative refinement procedure is performed again. If two successive values of the Euclidean norm of X1 show the norms are increasing, the iterative refinement procedure is terminated.

2. USAGE

a. Calling Sequence

CALL SOLVER (A,ADUM,U,N,NA,B,DBUM,V,M,NB,C,CDUM,NC,X,WORK1,EPSA,

EPSB,XDUM,TOL,ITER)

b. Input arguments

A,B,C   Matrices packed by columns in one-dimensional arrays. Not destroyed upon return. Dimensions are A - N*N, B - M*M, C - N*M.

ADUM,
BDUM,
CDUM,
WORK1,
XDUM,
U,V   Working space vectors of dimensions at least, ADUM - NA*NA, BDUM - NB*NB, CDUM - NC*NC, XDUM - N*M, WORK1 - 2*(N*M), U - N*N, V - M*M

ITER   Integer scalar indicating the maximum number of iterative refinements attempted by the subroutine. If ITER .LT. 0, no refinement iterations are performed.

TOL   Real scalar indicating the tolerance value used to determine if the norm of the equation error matrix is sufficiently small.

N,M,NA,
NB,NC   Integer scalars indicating matrix dimensions. It is required that NA .GT. N+1, NB .GT. M+1, and NC .GT. (MAX(N,M)+1).

EPSA,
EPSB   See AXBMXC subroutine description.

c. Output Arguments

X Upon normal return, X contains the solution to Eq. 1 packed by columns in a one-dimensional array.

d. COMMON Blocks

None

e. Error Messages

If the number of iterative refinements exceeds ITER or the iterative refinement proceed is unstable, the message "SYSTEM FAILED TO CONVERGE AFTER __________ ITERATIONS. RESIDUAL NORM = __________" is printed.

f. Subroutines Employed by SOLVER

ORACLS - ADD, NORMS, BCKMLT, HSHLDR, SCHUR, SYSSLV

PIFLIB - AXBMXC, SHRSOL

g. Subroutines Employing SOLVER

PIFLIB - CGTPIF

h. Conclusing Remarks

This subroutine is based on the algorithm reported in Ref. 3.

IV. SUMMARY - DESIGN PROCEDURE

The design procedure for determining a PIF autopilot control law using the program PIFCGT is outlined below:

Choose the control law sampling interval- DELT

Specify convergence criteria- EPS, ECGT, TOL, ELOG

Specify number of iterative steps- ITER MLOG, IKDGM

Choose autopilot design- ICHOSE

Specify ICH logicals

Specify DEBUG logicals for print out

Specify ICGT for feedforward matrix construction

Determine the new values for the aerodynamic coefficients and reprogram AEROD accordingly

Determine the BETA HOLD feedforward matrices for a number of different forward velocities. Compute a straight line approximation to the ϕ_c to δ_r feedforward matrix element in S22 using forward velocity as the independent variable. Reprogram HDCON based on the new straight line approximation.

Choose the number of simulation time steps - IMAX

Choose the simulation command magnitude - UMAG

Choose the starting and stopping time for the simulation command - USTRT, USTP

Specify the eigenvalue eigenvector printout- IFREEF, IFREEP

Choose stabilization criteria - DSTAB, DREAL

Choose the design flight condition by specifying forward velocity- VTP

Guess at the quadratic cost weighting matrices

Guess at the command model design parameters - GKDGM, GKPSI, ZDDMAX, ZCLOSE, ZDOMAX, AK1, AK2, ZCLMAX, ZCLMIN, ZCDMAX, GKMIN, GKDSLP, GKDICP, GKDMAX

Iterate at guessing command model design parameters until desired command model response is obtained

Iterate at guessing cost function weighting matrices until a desired tradeoff is made between gain magnitudes, command response and closed-loop eigenvalues

If the NAMELISTS in Appendix A are used, the only choices that should be required by the designer are for DELT, VTP, and the quadratic weights. AEROD and HDCON need not be modified if the NAVION aircraft is used for control design.

REFERENCES

1. Broussard, J. R., "Design, Implementation and Flight Testing of PIF Autopilot Designs for General Aviation Aircraft", NASA CR-3709, 1983.

2. Armstrong, E. S., ORACLS - A Design System for Linear Multivariable Control, Marcel - Dekher, Inc., New York, 1980. ORACLS can be obtained from the NASA sponsored Computer Software Management and Information Center (COSMIC), Suite 12, Barrow Hall, University of Georgia, Athens, GA 30602, by requesting program LAR-12313.

3. Barraud, A. Y., "A Numerical Algorithm to Solve $A^TXA-X=Q$", IEEE Trans. Automatic Control, Vol. AC-22, No. 3, October 1977, pp. 883-885.

4. Broussard, J. R., and O'Brien, M. J., "Feedforward Control to Track the Output of a Forcem Model", IEEE Trans. Automatic Control, Vol. AC-25, August 1980, pp. 851-852. A longer version appears in 17th IEEE Conference on Decision and Control, San Diego, California, January 1979.

5. Maybeck, P. S., Stochastic Models, Estimation, and Control, Vol., 4, Academic Press, New York, 1982.

6. Broussard, J. R., and Berry, P. W., "The Relationship Between Implicit Model Following and Eigenvalue/Eigenvector Placement", IEEE Trans. Automatic Control, Vol. AC-23, February 1978, pp. 79-81.

7. CYBER LOADER VERSION 1 REFERENCE MANUAL, Control Data Corporation (CDC), Publication No. 60429800, 215 Moffett Park Drive, Sunnyvale, California 94086.

APPENDIX A

NAMELISTS FOR AUTOPILOT DESIGNS

This appendix presents the nine NAMELISTS used by PIFCGT to design the autopilots in Ref. 1. Initial data values, gain values and quadratic weights are identified in the NAMELISTS.

PIF CONTROL DESIGN FOR BETA HOLD AUTOPILOT

```
$NAM1
  N=9,            M=4,             L=4,             NM=2,
  MM=2,
  ICHOSE=1,       IKDGM=1,         GKDGM=0.0020,    GKPSI=10.00,
  ZDDMAX=5.0,     ZCLOSE=9.00,     ZDOMAX=15.0,     AK1=1.0,
  AK2=2.00,       ZCLMAX=0.3,      ZCLMIN=0.1,      ZCDMAX=0.8,
  ICH=250*.T.,
  ICH(1,4)=.T.,   ICH(1,5)=.T.,    ICH(1,6)=.F.,    ICH(1,7)=.F.,
  ICH(1,8)=.F.,   ICH(1,9)=.F.,    ICH(1,10)=.T.,   ICH(1,11)=.T.,
  ICH(1,12)=.F.,  ICH(1,13)=.F.,   ICH(1,14)=.F.,   ICH(1,16)=.T.,
  ICH(2,2)=.T.,   ICH(2,3)=.F.,    ICH(2,4)=.T.,    ICH(2,5)=.T.,
  ICH(2,7)=.T.,   ICH(2,8)=.T.,    ICH(2,9)=.T.,    ICH(2,10)=.T.,
  ICH(2,12)=.F.,  ICH(2,13)=.T.,   ICH(2,14)=.F.,   ICH(2,15)=.F.,
  ICH(2,11)=.T.,  ICH(2,16)=.F.,
  ICH(4,2)=.T.,   ICH(4,3)=.F.,    ICH(4,4)=.T.,    ICH(4,5)=.T.,
  ICH(4,6)=.T.,   ICH(4,7)=.T.,    ICH(4,8)=.T.,    ICH(4,9)=.T.,
  ICH(4,10)=.F.,  ICH(4,11)=.F.,   ICH(4,15)=.T.,   ICH(4,16)=.F.,
  ICH(4,17)=.F.,
  ICH(5,1)=.F.,   ICH(5,2)=.F.,    ICH(5,3)=.T.,    ICH(5,4)=.T.,
  ICH(5,5)=.T.,   ICH(5,6)=.T.,
  SS=1.0,         EPS=1.0E-12,     ECGT=1.0E-12,    TOL=1.0E-12,
  ITEP=12,        VTP=175.0,
  ICGT=5*1,       ICGT(1)=3,       ICGT(4)=2,       ICGT(5)=0,
  ELOG=1.0E-10,   MLOG=7,          DREAL=0.90,      IDREG=5*0,
$
$NAM2
  XR=12*.F.,      UR=4*.F.,        YR=4*.F.,        XMR=12*.F.,
  UMR=4*.F.,
$
$NAM3
  IMAX=192,       IRUN=1,          UMAG=16*0.0,     USTRT=16*0.0,
  USTP=16*0.0,    UMAG(1,1)=20.,   USTRT(1,1)=1.0,USTP(1,1)=20.0,
$
$NAM4
  IFREEF=0,       IFREIG=0,        IFREEP=2,        DELT=0.1,
  DEBUM1=.F.,     DEBUM2=.F.,      DEBUM3=.F.,      DEBUM4=.F.,
  DEBUM6=.F.,     DEBU14=.F.,      DEBUMA=.F.,      DEBU22=.T.,
  DEBU24=.F.,     DEBU25=.F.,      DEBU26=.F.,      DEBU27=.F.,
  DEBU31=.F.,     DEBU42=.F.,      DEBU43=.F.,      METERS=.F.,
  DEBU44=.F.,     DEBU46=.F.,      DEBU48=.T.,      DEB416=.T.,
  DEBU49=.T.,     DEBU53=.T.,      DEBU54=.F.,
$
```

```
$NAM5
  F=81*0.0,          G=36*0.0,          H=48*0.0,          D=16*0.0,
  FM=4*0.0,          GM=4*0.0,          HM=8*0.0,          DM=8*0.0,
$
$NAM6
  UBQ=0.00E-00,      WBQ=0.00E-00,      QBQ=0.00E-00,      THQ= 0.00E-00,
  XIQ=0.0,           ZIQ=0.0,           VHQ=0.00E-00,      RBQ= 11.0E-00,
  PBQ=0.00E-00,      PHQ=10.0E-00,      PSQ=10.0E-00,      YIQ= 0.00E-00,
  DTH=0.00E-00,      DER=0.00E-00,      DA=0.10E-00,       DRR= 0.10E-00,
  DTHD=0.0E-00,      DERD=0.00E-00,     DARD=7.00E-00,     DRRD=7.00E-00,
  Z1Q=0.00E-00,      Z2Q=0.00E-00,      Z3Q=3.00E-00,      Z4Q= 2.50E-00,
  UBDQ=0.00E-00,     WBDQ=0.00E-00,     QBDQ=0.00E-00,     THDQ=0.00E-00,
  XIDQ=0.00E-00,     ZIDQ=0.00E-00,     VBDQ=0.00E-00,     RBDQ=0.00E-00,
  PBDQ=0.00E-00,     PHDQ=0.00E-00,     PSDQ=0.00E-00,     YIDQ=0.0,
$
$AERO1
$
$AERO2
$
$NAM7
  EI=9*0.0,          ER=9*0.0,          EI=9*0.0,
  ERV=400*0.0,       IEV=9*0,
$
```

```
PIF CONTROL DESIGN FOR ROLL SEL AUTOPILOT
$NAM1
  N=9,            M=4,            L=4,            NM=2,
  MM=2,
  ICHOSE=12,      IKDGM=1,        GKDGM=0.0020,   GKPSI=10.00,
  ZDDMAX=20.,     ZCLOSE=3.00,    ZDOMAX=10.0,    AK1=6.0,
  AK2=9.00,       ZCLMAX=0.3,     ZCLMIN=0.1,     ZCDMAX=0.8,
  ICH=250*.T.,
  ICH(1,4)=.T.,   ICH(1,5)=.T.,   ICH(1,6)=.F.,   ICH(1,7)=.F.,
  ICH(1,8)=.F.,   ICH(1,9)=.F.,   ICH(1,10)=.T.,  ICH(1,11)=.T.,
  ICH(1,12)=.F.,  ICH(1,13)=.F.,  ICH(1,14)=.F.,  ICH(1,16)=.T.,
  ICH(2,2)=.T.,   ICH(2,3)=.F.,   ICH(2,4)=.T.,   ICH(2,5)=.T.,
  ICH(2,7)=.T.,   ICH(2,8)=.T.,   ICH(2,9)=.T.,   ICH(2,10)=.T.,
  ICH(2,12)=.F.,  ICH(2,13)=.F.,  ICH(2,14)=.F.,  ICH(2,15)=.F.,
  ICH(2,11)=.T.,  ICH(2,16)=.F.,
  ICH(4,2)=.T.,   ICH(4,3)=.F.,   ICH(4,4)=.T.,   ICH(4,5)=.T.,
  ICH(4,6)=.T.,   ICH(4,7)=.T.,   ICH(4,8)=.T.,   ICH(4,9)=.T.,
  ICH(4,10)=.F.,  ICH(4,11)=.F.,  ICH(4,15)=.T.,  ICH(4,16)=.T.,
  ICH(5,1)=.F.,   ICH(5,2)=.F.,   ICH(5,3)=.T.,   ICH(5,4)=.T.,
  ICH(5,5)=.T.,   ICH(5,6)=.T.,
  SS=1.0,         EPS=1.0E-10,    ECGT=1.0E-10,   TOL=1.0E-10,
  ITER=10,        VTP=145.0,
  ICGT=5*1,       ICGT(1)=3,      ICGT(4)=0,      ICGT(5)=0,
  ELOG=1.0E-10,   MLOG=7,         DREAL=0.90,     IDREG=5*0,
$
$NAM2
  XR=12*.F.,      UR=4*.F.,       YR=4*.F.,       XMR=12*.F.,
  UMR=4*.F.,
$
$NAM3
  IMAX=152,       IRUN=1,         UMAG=16*0.0,    USTRT=16*0.0,
  USTP=16*0.0,    UMAG(1,1)=5.0,  USTRT(1,1)=1.0,USTP(1,1)=20.0,
$
$NAM4
  IFREEF=0,       IFREIG=0,       IFREEP=2,       DELT=0.1,
  DEBUM1=.F.,     DEBUM2=.F.,     DEBUM3=.F.,     DEBUM4=.F.,
  DEBUM6=.F.,     DEBU14=.F.,     DEBUMA=.F.,     DEBU22=.T.,
  DEBU24=.F.,     DEBU25=.T.,     DEBU26=.F.,     DEBU27=.F.,
  DEBU31=.F.,     DEBU42=.F.,     DEBU43=.F.,     METERS=.T.,
  DEBU44=.F.,     DEBU46=.F.,     DEBU48=.F.,     DEB416=.T.,
  DEBU49=.T.,     DEBU53=.T.,     DEBU54=.F.,
$
```

```
$NAM5
  F=81*0.0,        G=36*0.0,        H=48*0.0,        D=16*0.0,
  FM=4*0.0,        GM=4*0.0,        HM=8*0.0,        DM=8*0.0,
$
$NAM6
  UBQ=0.00E-00,    WBQ=0.00E-00,    QBQ=0.00E-00,    THQ=0.00E-00,
  XIQ=0.0,         ZIQ=0.0,         VHQ=0.00E-00,    RBQ=8.00E-00,
  PBQ=0.00E-00,    PHQ=6.00E-00,    PSQ=4.00E-00,    YIQ=0.00E-00,
  DTH=0.00E-00,    DER=0.00E-00,    DA=3.50E-00,     DRR=1.80E-00,
  DTHD=0.0E-00,    DERD=0.00E-00,   DARD=3.50E-00,   DRRD=4.00E-00,
  Z1Q=0.00E-00,    Z2Q=0.00E-00,    Z3Q=3.00E-00,    Z4Q=2.50E-00,
  UBDQ=0.00E-00,   WBDQ=0.00E-00,   QBDQ=0.00E-00,   THDQ=0.00E-00,
  XIDQ=0.00E-00,   ZIDQ=0.00E-00,   VBDQ=0.20E-00,   RBDQ=0.00E-00,
  PBDQ=0.00E-00,   PHDQ=0.00E-00,   PSDQ=0.00E-00,   YIDQ=0.0,
$
$AERO1
$
$AERO2
$
$NAM7
  EI=9*0.0,        ER=9*0.0,        EI=9*0.0,
  ERV=400*0.0,     IEV=9*0,
$
```

```
PIF CONTROL DESIGN FOR HDG SEL AUTOPILOT
$NAM1
  N=9,            M=4,            L=4,            NM=2,
  MM=2,
  ICHOSE=6,       IKDGM=1,        GKDGM=0.0020,   GKPSI=10.00,
  ZDDMAX=5.0,     ZCLOSE=9.00,    ZDOMAX=15.0,    AK1= 1.0,
  AK2=2.00,       ZCLMAX=0.3,     ZCLMIN=0.1,     ZCDMAX=0.8,
  ZCDMAX = 0.8,
  ICH=250*.T.,
  ICH(1,4)=.T.,   ICH(1,5)=.T.,   ICH(1,6)=.F.,   ICH(1,7)=.F.,
  ICH(1,8)=.F.,   ICH(1,9)=.F.,   ICH(1,10)=.T.,  ICH(1,11)=.T.,
  ICH(1,12)=.F.,  ICH(1,13)=.F.,  ICH(1,14)=.F.,  ICH(1,15)=.T.,
  ICH(1,16)=.T.,
  ICH(2,2)=.T.,   ICH(2,3)=.F.,   ICH(2,4)=.T.,   ICH(2,5)=.T.,
  ICH(2,7)=.T.,   ICH(2,8)=.T.,   ICH(2,9)=.T.,   ICH(2,10)=.T.,
  ICH(2,12)=.F.,  ICH(2,13)=.F.,  ICH(2,14)=.F.,  ICH(2,15)=.F.,
  ICH(2,11)=.T.,  ICH(2,16)=.F.,
  ICH(4,2)=.T.,   ICH(4,3)=.F.,   ICH(4,4)=.T.,   ICH(4,5)=.T.,
  ICH(4,6)=.T.,   ICH(4,7)=.T.,   ICH(4,8)=.T.,   ICH(4,9)=.T.,
  ICH(4,10)=.F.,  ICH(4,11)=.F.,
  ICH(5,1)=.F.,   ICH(5,2)=.F.,   ICH(5,3)=.T.,   ICH(5,4)=.T.,
  ICH(5,5)=.T.,   ICH(5,6)=.T.,
  SS=1.0,         EPS=1.0E-12,    ECGT=1.0E-12,   TOL=1.0E-12,
  ITER=12,        VTP=145.0,
  ICGT=5*1,       ICGT(1)=3,      ICGT(4)=0,      ICGT(5)=0,
  ELOG=1.0E-10,   MLOG=7,         DREAL=0.90,     IDREG=5*0,
$
$NAM2
  XR=12*.F.,      UR=4*.F.,       YR=4*.F.,       XMR=12*.F.,
  UMR=4*.F.,
$
$NAM3
  IMAX=292,       IRUN=1,         UMAG=16*0.0,    USTRT=16*0.0,
  USTP=16*0.0,    UMAG(1,1)=-45., USTRT(1,1)=0.5,USTP(1,1)=30.0,
$
$NAM4
  IFREEF=2,       IFREIG=0,       IFREEP=2,       DELT=0.1,
  DEBUM1=.F.,     DEBUM2=.F.,     DEBUM3=.F.,     DEBUM4=.F.,
  DEBUM6=.F.,     DEBU14=.F.,     DEBUMA=.F.,     DEBU22=.F.,
  DEBU24=.F.,     DEBU25=.F.,     DEBU26=.F.,     DEBU27=.F.,
  DEBU31=.F.,     DEBU42=.F.,     DEBU43=.F.,     METERS=.T.,
  DEBU44=.F.,     DEBU46=.F.,     DEBU48=.T.,     DEB416=.T.,
  DEBU49=.F.,     DEBU53=.T.,     DEBU54=.F.,
$
```

```
$NAM5
  F=81*0.0,       G=36*0.0,       H=48*0.0,       D=16*0.0,
  FM=4*0.0,       GM=4*0.0,       HM=8*0.0,       DM=8*0.0,
$
$NAM6
  UBQ=0.00E-00,  WBQ=0.00E-00,  QBQ=0.00E-00,  THQ=0.00E-00,
  XIQ=0.0,       ZIQ=0.0,       VHQ=0.00E-00,  RBQ=11.0E-00,
  PBQ=0.00E-00,  PHQ=10.0E-00,  PSQ=10.0E-00,  YIQ=0.00E-00,
  DTH=0.00E-00,  DER=0.00E-00,  DA=0.10E-00,   DRR=0.10E-00,
  DTHD=0.0E-00,  DERD=0.00E-00, DARD=7.00E-00, DRRD=7.00E-00,
  Z1Q=0.00E-00,  Z2Q=0.00E-00,  Z3Q=3.00E-00,  Z4Q=2.50E-00,
  UBDQ=0.00E-00, WBDQ=0.00E-00, QBDQ=0.00E-00, THDQ=0.00E-00,
  XIDQ=0.00E-00, ZIDQ=0.00E-00, VBDQ=0.00E-00, RBDQ=0.00E-00,
  PBDQ=0.00E-00, PHDQ=0.00E-00, PSDQ=0.00E-00, YIDQ=0.0,
$
$AERO1
$
$AERO2
$
$NAM7
  EI=9*0.0,      ER=9*0.0,       EI=9*0.0,
  ERV=400*0.0,   IEV=9*0,
$
```

```
PIF CONTROL DESIGN FOR APR LOCI AUTOPILOT
$NAM1
  N=9,              M=4,              L=4,              NM=2,
  MM=2,
  ICHOSE=9,         IKDGM=10,         GKDGM=0.0025,     GKPSI=5.000,
  ZDDMAX=10.,       ZCLOSE=15.0,      ZDOMAX=360.,      AK1=0.114,
  AK2=1.20,         ZCLMAX=0.3,       ZCLMIN=0.1,       ZCDMAX=0.5,
  GKDMIN = 0.11,    GKDSLP=-0.0020,   GKDICP=0.20,      GKDMAX=0.22,
  ICH=250*.T.,
  ICH(1,4)=.T.,     ICH(1,5)=.T.,     ICH(1,6)=.F.,     ICH(1,7)=.F.,
  ICH(1,8)=.F.,     ICH(1,9)=.F.,     ICH(1,10)=.T.,    ICH(1,11)=.T.,
  ICH(1,12)=.F.,    ICH(1,13)=.F.,    ICH(1,14)=.F.,    ICH(1,16)=.T.,
  ICH(2,2)=.T.,     ICH(2,3)=.F.,     ICH(2,4)=.T.,     ICH(2,5)=.T.,
  ICH(2,7)=.T.,     ICH(2,8)=.T.,     ICH(2,9)=.T.,     ICH(2,10)=.T.,
  ICH(2,11)=.T.,    ICH(2,12)=.F.,    ICH(2,13)=.F.,    ICH(2,14)=.F.,
  ICH(2,15)=.F.,    ICH(2,16)=.F.,
  ICH(4,2)=.T.,     ICH(4,3)=.F.,     ICH(4,4)=.T.,     ICH(4,5)=.T.,
  ICH(4,6)=.T.,     ICH(4,7)=.T.,     ICH(4,8)=.T.,     ICH(4,9)=.T.,
  ICH(4,10)=.F.,    ICH(4,11)=.F.,    ICH(4,15)=.T.,      ICH(4,16)=.T.,
  ICH(5,1)=.F.,     ICH(5,2)=.F.,     ICH(5,3)=.T.,     ICH(5,4)=.T.,
  ICH(5,5)=.T.,     ICH(5,6)=.T.,
  SS=1.0,           EPS=1.0E-10,      ECGT=1.0E-10,     TOL=1.0E-10,
  ITER=10,          VTP=145.0,
  ICGT=5*1,         ICGT(1)=3,        ICGT(4)=0,        ICGT(5)=0,
  ELOG=1.0E-10,     MLOG=7,           DREAL=0.90,       IDREG=5*0,
$
$NAM2
  XR=12*.F.,        UR=4*.F.,         YR=4*.F.,         XMR=12*.F.,
  UMR=4*.F.,
$
$NAM3
  IMAX=348,         IRUN=1,           UMAG=16*0.0,      USTRT=16*0.0,
  USTP=16*0.0,      UMAG(1,1)=500.,USTRT(1,1)=0.0,USTP(1,1)=30.0,
  UMAG(2,1)=45.,    USTRT(2,1)=0.0,USTP(2,1)=30.0,
$
$NAM4
  IFREEF=0,         IFREIG=0,         IFREEP=1,         DELT=0.1,
  DEBUM1=.F.,       DEBUM2=.F.,       DEBUM3=.F.,       DEBUM4=.F.,
  DEBUM6=.F.,       DEBU14=.F.,       DEBUMA=.F.,       DEBU22=.T.,
  DEBU24=.F.,       DEBU25=.T.,       DEBU26=.F.,       DEBU27=.F.,
  DEBU31=.F.,       DEBU42=.F.,       DEBU43=.F.,       METERS=.T.,
  DEBU44=.F.,       DEBU46=.F.,       DEBU48=.F.,       DEB416=.T.,
  DEBU49=.T.,       DEBU53=.T.,       DEBU54=.F.,
$
```

```
$NAM5
  F=81*0.0,          G=36*0.0,        H=48*0.0,        D=16*0.0,
  FM=4*0.0,          GM=4*0.0,        HM=8*0.0,        DM=8*0.0,
$
$NAM6
  UBQ=0.00E-00,   WBQ=0.00E-00,   QBQ=0.00E-00,   THQ=0.00E-00,
  XIQ=0.0,        ZIQ=0.0,        VHQ=0.00E-00,   RBQ=11.0E-00,
  PBQ=0.00E-00,   PHQ=10.0E-00,   PSQ=14.0E-00,   YIQ=0.00E-00,
  DTH=0.00E-00,   DER=0.00E-00,   DA=0.10E-00,    DRR=0.10E-00,
  DTHD=0.0E-00,   DERD=0.00E-00,  DARD=7.00E-00,  DRRD=7.00E-00,
  Z1Q=0.00E-00,   Z2Q=0.00E-00,   Z3Q=2.50E-00,   Z4Q=0.50E-00,
  UBDQ=0.00E-00,  WBDQ=0.00E-00,  QBDQ=0.00E-00,  THDQ=0.00E-00,
  XIDQ=0.00E-00,  ZIDQ=0.00E-00,  VBDQ=0.00E-00,  RBDQ=0.00E-00,
  PBDQ=0.00E-00,  PHDQ=0.00E-00,  PSDQ=0.00E-00,  YIDQ=0.0,
$
$AERO1
$
$AERO2
$
$NAM7
  EI=9*0.0,        ER=9*0.0,        EI=9*0.0,
  ERV=400*0.0,     IEV=9*0,
$
```

```
PIF CONTROL DESIGN FOR APR LOCR AUTOPILOT
$NAM1
  N=9,             M=4,             L=4,             NM=2,
  MM=2,
  ICHOSE=15,       IKDGM=1,         GKDGM=0.0020,    GKPSI=3.00,
  ZDOMAX=10.,      ZCLOSE=15.0,     ZDOMAX=360.,     AK1=0.114,
  AK2=1.20,        ZCLMAX=0.3,      ZCLMIN=0.1,      ZCDMAX=0.5,
  GKDMIN = 0.11,   GKDSLP=-0.0020,GKDICP=0.20,       GKDMAX=0.22,
  ICH=250*.T.,     GKY=0.30,
  ICH(1,4)=.T.,    ICH(1,5)=.T.,    ICH(1,6)=.F.,    ICH(1,7)=.F.,
  ICH(1,8)=.F.,    ICH(1,9)=.F.,    ICH(1,10)=.T.,   ICH(1,11)=.F.,
  ICH(1,12)=.F.,   ICH(1,13)=.F.,   ICH(1,14)=.F.,   ICH(1,16)=.T.,
  ICH(2,2)=.T.,    ICH(2,3)=.F.,    ICH(2,4)=.T.,    ICH(2,5)=.T.,
  ICH(2,7)=.T.,    ICH(2,8)=.T.,    ICH(2,9)=.T.,    ICH(2,10)=.T.,
  ICH(2,12)=.T.,   ICH(2,13)=.F.,   ICH(2,14)=.F.,   ICH(2,15)=.F.,
  ICH(2,11)=.T.,   ICH(2,16)=.F.,
  ICH(4,2)=.T.,    ICH(4,3)=.F.,    ICH(4,4)=.T.,    ICH(4,5)=.T.,
  ICH(4,6)=.T.,    ICH(4,7)=.T.,    ICH(4,8)=.T.,    ICH(4,9)=.T.,
  ICH(4,10)=.F.,   ICH(4,11)=.F.,   ICH(4,15)=.T.,     ICH(4,16)=.T.,
  ICH(5,1)=.F.,    ICH(5,2)=.F.,    ICH(5,3)=.T.,    ICH(5,4)=.T.,
  ICH(5,5)=.T.,    ICH(5,6)=.T.,
  SS=1.0,          EPS=1.0E-10,     ECGT=1.0E-10,    TOL=1.0E-10,
  ITER=10,         VTP=145.0,
  ICGT=5*1,        ICGT(1)=3,       ICGT(4)=0,       ICGT(5)=0,
  ELOG=1.0E-10,    MLOG=7,          DREAL=0.90,      IDREG=5*0,
$
$NAM2
  XR=12*.F.,       UR=4*.F.,        YR=4*.F.,        XMR=12*.F.,
  UMR=4*.F.,
$
$NAM3
  IMAX=346,        IRUN=1,          UMAG=16*0.0,     USTRT=16*0.0,
  USTP=16*0.0,     UMAG(1,1)=500.,USTRT(1,1)=0.0,USTP(1,1)=30.0,
  UMAG(2,1)=45.,   USTRT(2,1)=0.0,USTP(2,1)=30.0,
$
$NAM4
  IFREEF=0,        IFREIG=0,        IFREEP=1,        DELT=0.1,
  DEBUM1=.F.,      DEBUM2=.F.,      DEBUM3=.F.,      DEBUM4=.F.,
  DEBUM6=.F.,      DEBU14=.F.,      DEBUMA=.F.,      DEBU22=.T.,
  DEBU24=.F.,      DEBU25=.F.,      DEBU26=.F.,      DEBU27=.F.,
  DEBU31=.F.,      DEBU42=.F.,      DEBU43=.F.,      METERS=.T.,
  DEBU44=.F.,      DEBU46=.F.,      DEBU48=.F.,      DEB416=.T.,
  DEBU49=.T.,      DEBU53=.T.,      DEBU54=.F.,
$
```

```
$NAM5
  F=81*0.0,        G=36*0.0,        H=48*0.0,        D=16*0.0,
  FM=4*0.0,        GM=4*0.0,        HM=8*0.0,        DM=8*0.0,
$
$NAM6
  UBQ=0.00E-00,    WBQ=0.00E-00,    QBQ=0.00E-00,    THQ=0.00E-00,
  XIQ=0.0,         ZIQ=0.0,         VHO=0.00E-00,    RBQ=11.0E-00,
  PBQ=0.00E-00,    PHQ=10.0E-00,    PSQ=11.0E-00,    YIQ=0.00E-00,
  DTH=0.00E-00,    DER=0.00E-00,    DA=0.10E-00,     DRR=0.10E-00,
  DTHD=0.0E-00,    DERD=0.00E-00,   DARD=7.00E-00,   DRRD=7.00E-00,
  Z1Q=0.00E-00,    Z2Q=0.00E-00,    Z3Q=1.00E-00,    Z4Q=2.20E-00,
  UBDQ=0.00E-00,   WBDQ=0.00E-00,   QBDQ=0.00E-00,   THDQ=0.00E-00,
  XIDQ=0.00E-00,   ZIDQ=0.00E-00,   VBDQ=0.00E-00,   RBDQ=0.00E-00,
  PBDQ=0.00E-00,   PHDQ=0.00E-00,   PSDQ=0.00E-00,   YIDQ=0.0,
$
$AERO1
$
$AERO2
$
$NAM7
  EI=9*0.0,        ER=9*0.0,        EI=9*0.0,
  ERV=400*0.0,     IEV=9*0,
$
```

```
PIF CONTROL DESIGN FOR APR LOCP AUTOPILOT
$NAM1
  N=9,            M=4,              L=4,              NM=2,
  MM=2,
  ICHOSE=16,      IKDGM=1,          GKDGM=0.0020,     GKPSI=4.00,
  ZDDMAX=10.,     ZCLOSE=15.0,      ZDOMAX=360.,      AK1=0.114,
  AK2=1.20,       ZCLMAX=0.3,       ZCLMIN=0.1,       ZCDMAX=0.5,
  ZCDMAX = 0.5,   XS=0.00,          YS=0.0,           ZS=-0.00,
  GKDMIN = 0.11,  GKDSLP=-0.0020,GKDICP=0.20,         GKDMAX=0.22,
  ICH=250*.T.,    GKY=0.70,
  ICH(1,4)=.T.,   ICH(1,5)=.T.,     ICH(1,6)=.F.,     ICH(1,7)=.F.,
  ICH(1,8)=.F.,   ICH(1,9)=.F.,     ICH(1,10)=.T.,    ICH(1,11)=.F.,
  ICH(1,12)=.F.,  ICH(1,13)=.F.,    ICH(1,14)=.F.,    ICH(1,16)=.T.,
  ICH(2,2)=.T.,   ICH(2,3)=.F.,     ICH(2,4)=.T.,     ICH(2,5)=.T.,
  ICH(2,7)=.T.,   ICH(2,8)=.T.,     ICH(2,9)=.T.,     ICH(2,10)=.T.,
  ICH(2,12)=.T.,  ICH(2,13)=.F.,    ICH(2,14)=.F.,    ICH(2,15)=.F.,
  ICH(2,11)=.T.,  ICH(2,16)=.F.,
  ICH(4,2)=.T.,   ICH(4,3)=.F.,     ICH(4,4)=.T.,     ICH(4,5)=.T.,
  ICH(4,6)=.T.,   ICH(4,7)=.T.,     ICH(4,8)=.T.,     ICH(4,9)=.T.,
  ICH(4,10)=.F.,  ICH(4,11)=.F.,    ICH(4,15)=.T.,      ICH(4,16)=.T.,
  ICH(5,1)=.F.,   ICH(5,2)=.F.,     ICH(5,3)=.T.,     ICH(5,4)=.T.,
  ICH(5,5)=.T.,   ICH(5,6)=.T.,
  SS=1.0,         EPS=1.0E-10,      ECGT=1.0E-10,     TOL=1.0E-10,
  ITER=10,        VTP=150.0,
  ICGT=5*1,       ICGT(1)=3,        ICGT(4)=0,        ICGT(5)=0,
  ELOG=1.0E-10,   MLOG=7,           DREAL=0.90,       IDREG=5*0,
$
$NAM2
  XR=12*.F.,      UR=4*.F.,         YR=4*.F.,         XMR=12*.F.,
  UMR=4*.F.,
$
$NAM3
  IMAX=346,       IRUN=1,           UMAG=16*0.0,      USTRT=16*0.0,
  USTP=16*0.0,    UMAG(1,1)=500.,USTRT(1,1)=0.0,USTP(1,1)=30.0,
  UMAG(2,1)=45.,  USTRT(2,1)=0.0,USTP(2,1)=30.0,
$
$NAM4
  IFREEF=0,       IFREIG=0,         IFREEP=1,         DELT=0.1,
  DEBUM1=.F.,     DEBUM2=.F.,       DEBUM3=.F.,       DEBUM4=.F.,
  DEBUM6=.F.,     DEBU14=.F.,       DEBUMA=.F.,       DEBU22=.T.,
  DEBU24=.F.,     DEBU25=.F.,       DEBU26=.F.,       DEBU27=.F.,
  DEBU31=.F.,     DEBU42=.F.,       DEBU43=.F.,       METERS=.T.,
  DEBU44=.F.,     DEBU46=.F.,       DEBU48=.F.,       DEB416=.T.,
  DEBU49=.T.,     DEBU53=.T.,       DEBU54=.F.,
$
```

```
$NAM5
  F=81*0.0,        G=36*0.0,        H=48*0.0,        D=16*0.0,
  FM=4*0.0,        GM=4*0.0,        HM=8*0.0,        DM=8*0.0,
$
$NAM6
  UBQ=0.00E-00,    WBQ=0.00E-00,    QBQ=0.00E-00,    THQ=0.00E-00,
  XIQ=0.0,         ZIQ=0.0,         VHQ=0.00E-00,    RBQ=11.0E-00,
  PBQ=0.00E-00,    PHQ=8.00E-00,    PSQ=14.0E-00,    YIQ=0.00E-00,
  DTH=0.00E-00,    DER=0.00E-00,    DA=0.10E-00,     DRR=0.10E-00,
  DTHD=0.0E-00,    DERD=0.00E-00,   DARD=7.00E-00,   DRRD=7.00E-00,
  Z1Q=0.00E-00,    Z2Q=0.00E-00,    Z3Q=1.00E-00,    Z4Q=2.20E-00,
  UBDQ=0.00E-00,   WBDQ=0.00E-00,   QBDQ=0.00E-00,   THDQ=0.00E-00,
  XIDQ=0.00E-00,   ZIDQ=0.00E-00,   VBDQ=0.00E-00,   RBDQ=0.00E-00,
  PBDQ=0.00E-00,   PHDQ=0.00E-00,   PSDQ=0.00E-00,   YIDQ=0.0,
$
$AERO1
$
$AERO2
$
$NAM7
  EI=9*0.0,        ER=9*0.0,        EI=9*0.0,
  ERV=400*0.0,     IEV=9*0,
$
```

```
PIF CONTROL DESIGN FOR PITCH SEL AUTOPILOT
$NAM1
  N=9,             M=4,              L=4,             NM=2,
  MM=2,
  ICHOSE=4,        IKDGM=1,          GKDGM=0.0020,    GKPSI=10.00,
  ZDDMAX=0.75,     ZCLOSE=19.68,     ZDOMAX=8.30,     AK1=0.8,
  AK2=2.00,        ZCLMAX=2.0,       ZCLMIN=0.4,      ZCDMAX=0.3,
  GKDMIN=0.04,   GKDSLP=-0.0004,GKDICP=0.07,     GKDMAX=0.07,
  ICH=250*.T.,
  ICH(1,4)=.T.,    ICH(1,5)=.T.,     ICH(1,6)=.F.,    ICH(1,7)=.F.,
  ICH(1,8)=.F.,    ICH(1,9)=.F.,     ICH(1,10)=.T.,   ICH(1,11)=.T.,
  ICH(1,12)=.F.,   ICH(1,13)=.F.,    ICH(1,14)=.F.,   ICH(1,15)=.T.,
  ICH(1,16)=.T.,
  ICH(2,2)=.T.,    ICH(2,3)=.F.,     ICH(2,4)=.T.,    ICH(2,5)=.T.,
  ICH(2,7)=.T.,    ICH(2,8)=.T.,     ICH(2,9)=.T.,    ICH(2,10)=.T.,
  ICH(2,12)=.T.,   ICH(2,13)=.F.,    ICH(2,14)=.F.,   ICH(2,15)=.F.,
  ICH(2,11)=.T.,   ICH(2,16)=.F.,
  ICH(4,2)=.F.,    ICH(4,3)=.T.,     ICH(4,4)=.T.,    ICH(4,5)=.T.,
  ICH(4,6)=.T.,    ICH(4,7)=.T.,     ICH(4,8)=.T.,    ICH(4,9)=.T.,
  ICH(4,10)=.F.,   ICH(4,11)=.F.,
  ICH(5,1)=.F.,    ICH(5,2)=.F.,     ICH(5,3)=.T.,    ICH(5,4)=.T.,
  ICH(5,5)=.T.,    ICH(5,6)=.T.,
  SS=1.0,          EPS=1.0E-10,      ECGT=1.0E-10,    TOL=1.0E-10,
  ITER=10,         VTP=150.0,
  ICGT=5*1,        ICGT(1)=3,        ICGT(4)=0,       ICGT(5)=0,
  ELOG=1.0E-10,    MLOG=7,           DREAL=0.90,      IDREG=5*0,
$
$NAM2
  XR=12*.F.,       UR=4*.F.,         YR=4*.F.,        XMR=12*.F.,
  UMR=4*.F.,
$
$NAM3
  IMAX=198,        IRUN=1,           UMAG=16*0.0,     USTRT=16*0.0,
  USTP=16*0.0,     UMAG(1,1)=5.00,USTRT(1,1)=1.0,USTP(1,1)=30.0,
$
$NAM4
  IFREEF=1,        IFREIG=0,         IFREEP=1,        DELT=0.1,
  DEBUM1=.F.,      DEBUM2=.F.,       DEBUM3=.F.,      DEBUM4=.F.,
  DEBUM6=.F.,      DEBU14=.F.,       DEBUMA=.F.,      DEBU22=.T.,
  DEBU24=.F.,      DEBU25=.F.,       DEBU26=.F.,      DEBU27=.F.,
  DEBU31=.F.,      DEBU42=.F.,       DEBU43=.T.,      METERS=.T.,
  DEBU44=.F.,      DEBU46=.F.,       DEBU48=.F.,      DEB416=.T.,
  DEBU49=.F.,      DEBU53=.T.,       DEBU54=.F.,
$
```

```
$NAM5
  F=81*0.0,         G=36*0.0,         H=48*0.0,         D=16*0.0,
  FM=4*0.0,         GM=4*0.0,         HM=8*0.0,         DM=8*0.0,
$
$NAM6
  UBQ=0.00E-00,     WBQ=0.00E-00,     QBQ=0.00E-00,     THQ=8.00E-00,
  XIQ=0.0,          ZIQ=0.5,          VHQ=0.00E-00,     RBQ=0.00E-00,
  PBQ=0.00E-00,     PHQ=0.00E-00,     PSQ=0.00E-00,     YIQ=0.00E-00,
  DTH=0.00E-00,     DER=0.00E-00,     DA=0.00E-00,      DRR=0.00E-00,
  DTHD=0.0E-00,     DERD=7.00E-00,    DARD=0.00E-00,    DRRD=0.00E-00,
  Z1Q=3.000E-00,    Z2Q=3.000E-00,    Z3Q=0.00E-00,     Z4Q=0.00E-00,
  UBDQ=0.00E-00,    WBDQ=0.00E-00,    QBDQ=0.00E-00,    THDQ=0.00E-00,
  XIDQ=0.00E-00,    ZIDQ=0.00E-00,    VBDQ=0.00E-00,    RBDQ=0.00E-00,
  PBDQ=0.00E-00,    PHDQ=0.00E-00,    PSDQ=0.00E-00,    YIDQ=0.0,
$
$AERO1
$
$AERO2
$
$NAM7
  EI=9*0.0,         ER=9*0.0,         EI=9*0.0,
  ERV=400*0.0,      IEV=9*0,
$
```

PIF CONTROL DESIGN FOR ALT SEL AUTOPILOT

```
$NAM1
  N=9,              M=4,              L=4,              NM=2,
  MM=2,
  ICHOSE=7,         IKDGM=1,          GKDGM=0.0020,     GKPSI=10.00,
  ZDDMAX=0.75,      ZCLOSE=19.68,     ZDOMAX=8.30,      AK1=0.8,
  AK2=2.00,         ZCLMAX=2.0,       ZCLMIN=0.4,       ZCDMAX=0.3,
  GKDMIN=0.04,      GKDSLP=-0.0004,GKDICP=0.07,         GKDMAX=0.07,
  ICH=250*.T.,
  ICH(1,4)=.T.,     ICH(1,5)=.T.,     ICH(1,6)=.F.,     ICH(1,7)=.F.,
  ICH(1,8)=.F.,     ICH(1,9)=.F.,     ICH(1,10)=.T.,    ICH(1,11)=.T.,
  ICH(1,12)=.F.,    ICH(1,13)=.F.,    ICH(1,14)=.F.,    ICH(1,15)=.T.,
  ICH(1,16)=.T.,
  ICH(2,2)=.T.,     ICH(2,3)=.F.,     ICH(2,4)=.T.,     ICH(2,5)=.T.,
  ICH(2,7)=.T.,     ICH(2,8)=.T.,     ICH(2,9)=.T.,     ICH(2,10)=.T.,
  ICH(2,12)=.T.,    ICH(2,13)=.F.,    ICH(2,14)=.F.,    ICH(2,15)=.F.,
  ICH(2,11)=.T.,    ICH(2,16)=.F.,
  ICH(4,2)=.F.,     ICH(4,3)=.T.,     ICH(4,4)=.T.,     ICH(4,5)=.T.,
  ICH(4,6)=.T.,     ICH(4,7)=.T.,     ICH(4,8)=.T.,     ICH(4,9)=.T.,
  ICH(4,10)=.F.,    ICH(4,11)=.F.,
  ICH(5,1)=.F.,     ICH(5,2)=.F.,     ICH(5,3)=.T.,     ICH(5,4)=.T.,
  ICH(5,5)=.T.,     ICH(5,6)=.T.,
  SS=1.0,           EPS=1.0E-10,      ECGT=1.0E-10,     TOL=1.0E-10,
  ITER=10,          VTP=150.0,
  ICGT=5*1,         ICGT(1)=3,        ICGT(4)=0,        ICGT(5)=0,
  ELOG=1.0E-10,     MLOG=7,           DREAL=0.90,       IDREG=5*0,
$
$NAM2
  XR=12*.F.,        UR=4*.F.,         YR=4*.F.,         XMR=12*.F.,
  UMR=4*.F.,
$
$NAM3
  IMAX=298,         IRUN=1,           UMAG=16*0.0,      USTRT=16*0.0,
  USTP=16*0.0,      UMAG(1,1)=100.,USTRT(1,1)=1.0,USTP(1,1)=30.0,
$
$NAM4
  IFREEF=1,         IFREIG=0,         IFREEP=1,         DELT=0.1,
  DEBUM1=.F.,       DEBUM2=.F.,       DEBUM3=.F.,       DEBUM4=.F.,
  DEBUM6=.F.,       DEBU14=.F.,       DEBUMA=.F.,       DEBU22=.T.,
  DEBU24=.F.,       DEBU25=.F.,       DEBU26=.F.,       DEBU27=.T.,
  DEBU31=.F.,       DEBU42=.F.,       DEBU43=.F.,       METERS=.T.,
  DEBU44=.T.,       DEBU46=.F.,       DEBU48=.F.,       DEB416=.T.,
  DEBU49=.F.,       DEBU53=.T.,       DEBU54=.F.,
$
```

```
$NAM5
  F=81*0.0,         G=36*0.0,         H=48*0.0,         D=16*0.0,
  FM=4*0.0,         GM=4*0.0,         HM=8*0.0,         DM=8*0.0,
$
$NAM6
  UBQ=0.00E-00,     WBQ=0.00E-00,     QBQ=0.00E-00,     THQ=14.0E-00,
  XIQ=0.0,          ZIQ=0.5,          VHQ=0.00E-00,     RBQ=0.00E-00,
  PBQ=0.00E-00,     PHQ=0.00E-00,     PSQ=0.00E-00,     YIQ=0.00E-00,
  DTH=0.00E-00,     DER=0.00E-00,     DA=0.00E-00,      DRR=0.00E-00,
  DTHD=0.0E-00,     DERD=7.00E-00,    DARD=0.00E-00,    DRRD=0.00E-00,
  Z1Q=0.00E-00,     Z2Q=0.25E-00,     Z3Q=0.00E-00,     Z4Q=0.00E-00,
  UBDQ=0.00E-00,    WBDQ=0.00E-00,    QBDQ=0.00E-00,    THDQ=0.00E-00,
  XIDQ=0.00E-00,    ZIDQ=1.00E-00,    VBDQ=0.00E-00,    RBDQ=0.00E-00,
  PBDQ=0.00E-00,    PHDQ=0.00E-00,    PSDQ=0.00E-00,    YIDQ=0.0,
$
$AERO1
$
$AERO2
$
$NAM7
  EI=9*0.0,         ER=9*0.0,         EI=9*0.0,
  ERV=400*0.0,      IEV=9*0,
$
```

```
PIF CONTROL DESIGN FOR APR GS AUTOPILOT
$NAM1
  N=9,              M=4,              L=4,              NM=2,
  MM=2,
  ICHOSE=14,        IKDGM=1,          GKDGM=0.0020,     GKPSI=10.00,
  ZDDMAX=0.03,      ZCLOSE=0.00,      ZDOMAX=0.00,      AK1=0.04,
  AK2=0.00,         ZCLMAX=0.0,       ZCLMIN=-5.0,      ZCDMAX=3.0,
  GKDMIN = 0.04, GKDSLP=-0.0004,GKDICP=0.07,      GKDMAX=0.07,
  ICH=250*.T.,
  ICH(1,4)=.T.,     ICH(1,5)=.T.,     ICH(1,6)=.F.,     ICH(1,7)=.F.,
  ICH(1,8)=.F.,     ICH(1,9)=.F.,     ICH(1,10)=.T.,    ICH(1,11)=.T.,
  ICH(1,12)=.F.,    ICH(1,13)=.F.,    ICH(1,14)=.F.,    ICH(1,16)=.T.,
  ICH(2,2)=.T.,     ICH(2,3)=.F.,     ICH(2,4)=.T.,     ICH(2,5)=.T.,
  ICH(2,7)=.T.,     ICH(2,8)=.T.,     ICH(2,9)=.T.,     ICH(2,10)=.T.,
  ICH(2,12)=.T.,    ICH(2,13)=.F.,    ICH(2,14)=.F.,    ICH(2,15)=.F.,
  ICH(2,11)=.T.,    ICH(2,16)=.F.,
  ICH(4,2)=.F.,     ICH(4,3)=.T.,     ICH(4,4)=.T.,     ICH(4,5)=.T.,
  ICH(4,6)=.T.,     ICH(4,7)=.T.,     ICH(4,8)=.T.,     ICH(4,9)=.T.,
  ICH(4,10)=.F.,    ICH(4,11)=.F.,    ICH(4,15)=.T.,    ICH(4,16)=.T.,
  ICH(5,1)=.F.,     ICH(5,2)=.F.,     ICH(5,3)=.T.,     ICH(5,4)=.T.,
  ICH(5,5)=.T.,     ICH(5,6)=.T.,
  SS=1.0,           EPS=1.0E-10,      ECGT=1.0E-10,     TOL=1.0E-10,
  ITER=10,          VTP=150.0,
  ICGT=5*1,         ICGT(1)=3,        ICGT(4)=0,        ICGT(5)=0,
  ELOG=1.0E-10,     MLOG=7,           DREAL=0.90,       IDREG=5*0,
$
$NAM2
  XR=12*.F.,        UR=4*.F.,         YR=4*.F.,         XMR=12*.F.,
  UMR=4*.F.,
$
$NAM3
  IMAX=292,         IRUN=1,           UMAG=16*0.0,      USTRT=16*0.0,
  USTP=16*0.0,      UMAG(1,1)=-100.,USTRT(1,1)=0.,USTP(1,1)=30.0,
$
$NAM4
  IFREEF=0,         IFREIG=0,         IFREEP=2,         DELT=0.1,
  DEBUM1=.F.,       DEBUM2=.F.,       DEBUM3=.F.,       DEBUM4=.F.,
  DEBUM6=.F.,       DEBU14=.F.,       DEBUMA=.F.,       DEBU22=.T.,
  DEBU24=.F.,       DEBU25=.T.,       DEBU26=.F.,       DEBU27=.F.,
  DEBU31=.F.,       DEBU42=.F.,       DEBU43=.F.,       METERS=.T.,
  DEBU44=.F.,       DEBU46=.F.,       DEBU48=.F.,       DEB416=.T.,
  DEBU49=.T.,       DEBU53=.T.,       DEBU54=.F.,
$
```

```
$NAM5
  F=81*0.0,         G=36*0.0,         H=48*0.0,         D=16*0.0,
  FM=4*0.0,         GM=4*0.0,         HM=8*0.0,         DM=8*0.0,
$
$NAM6
  UBQ=0.00E-00,   WBQ=0.00E-00,   QBQ=0.00E-00,   THQ=12.5E-00,
  XIQ=0.0,        ZIQ=0.5,        VHQ=0.00E-00,   RBQ=0.00E-00,
  PBQ=0.00E-00,   PHQ=0.00E-00,   PSQ=0.00E-00,   YIQ=0.00E-00,
  DTH=0.00E-00,   DER=0.00E-00,   DA=0.00E-00,    DRR=0.00E-00,
  DTHD=0.0E-00,   DERD=7.00E-00,  DARD=0.00E-00,  DRRD=0.00E-00,
  Z1Q=0.00E-00,   Z2Q=0.25E-00,   Z3Q=0.00E-00,   Z4Q=0.00E-00,
  UBDQ=0.00E-00,  HBDQ=0.00E-00,  QBDQ=0.00E-00,  THDQ=0.00E-00,
  XIDQ=0.00E-00,  ZIDQ=1.00E-00,  VBDQ=0.00E-00,  RBDQ=0.00E-00,
  PBDQ=0.00E-00,  PHDQ=0.00E-00,  PSDQ=0.00E-00,  YIDQ=0.0,
$
$AERO1
  LAMDA=0.0,      ZETA=0.0,      LAMDAGS=3.0,
$
$AERO2
  LAMDA1=0.0,     ZETA1=0.0,
$
$NAM7
  EI=9*0.0,       EP=9*0.0,       EI=9*0.0,
  ERV=400*0.0,    IEV=9*0,
$
```

APPENDIX B

SUBMIT FILE FOR AUTOPILOT DESIGNS

The following is a typical submit file containing the JOB CONTROL CARDS to execute the PIFCGT program on the NASA Langley Computer Complex using the NOS. 1.3 operating system (b̸ is a blank space).

```
/JOB
PIFNAM7,T100,CM160000.b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸b̸BIN15DT  DBTAYLOR
USER,XXXXXX.
CHARGE,XXXXXX,LRC.
GET,TAPE20=PIFNAM7.
GET,PIFLIBO/UN=1003675C.
GET,ORACLIB/UN=017545N.
MAP,OFF.
LDSET,LIB=PIFLIBO.
LDSET,USEP=AEROD.
LDSET,LIB=ORACLIB.
LDSET,PRESETA=NGINF.
PIFCGT.
BOMB.
EXIT.
ROUTE,OUTPUT,DC=PR,ID,DEF.
SRUCOMP.
REWIND,OUTPUT.
COPY,OUTPUT,TAPE86
```

```
REWIND,TAPE86.

REPLACE,TAPE86.

BOMB.

EXIT.

DAYFILE,BADP.

REPLACE,BADP.

EXIT.

/EOR

bPIFbCONTROLbDESIGNbPROGRAM

/EOF
```

To use the subroutine FREEF, the statements

```
ATTACH,FTNMLIB/UN=LIBRARY,NA.

LDSET,LIB=FTNMLIB
```

must be included. To use the subroutine DPIFSP for plotting, the statements

```
GET,ICSLIB/UN=004675C.

LDSET,LIB=ICSLIB.

REPLACE,TAPE1=PIFPLT7,
```

must be included in appropriate locations.

APPENDIX C

SEGMENTATION DIRECTIVES

PIFCGT is a large TOP-DOWN program. Central memory would be inefficiently used if the entire program had to be loaded for the program to execute. With segmentation, only certain portions of the executing program need to be in central memory concurrently. Segmented loading is initiated by the execution of a SEGLOAD control statement that causes SEGLOAD directives to be processed as discussed in Ref. 7. The SEGLOAD directives for PIFCGT are (ƀ is a blank space).

```
ƀƀINCLUDEƀAEROD
ƀƀTREE PIFCGT-(DIMSS,HDCON,CFWM,PIFMODL,PIFPLC,REMAT,DISCMOD, QRMP1,
ORMHAT,DREG,PIFG,DPIFS,DPIFSP,FGAERO,PIFEIG,RUNINFO,INTERP,PIFFRE)
PIFCGTƀGLOBALƀDISCSV,WORKS9,WGHTS,MODF,DISCM,PLA,DIMN1,DIMN2,DIMN3,
DIMN4,DIMN5,SMAT,PFIG,QWHT,EIGEN,LABEL,GLAD,LINES,IDENTS,AERO,FORM
TOL,CONV,SLVBLK,UDIR,DEBG,EIGS,HDXA,AER1,RNINFO,RNINFO1,RNINFO2,
STP.END,FCL.C.,Q8.IO.,FCL=ENT,FDL.COM
ƀEND
```

Alternate forms of memory allocation may have to be developed by a user depending on machine configuration.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-166123		

4. Title and Subtitle	5. Report Date
PIFCGT - A PIF AUTOPILOT DESIGN PROGRAM FOR GENERAL AVIATION AIRCRAFT	December 1983
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
John R. Broussard	TR-681103
	10. Work Unit No.
9. Performing Organization Name and Address Information & Control Systems, Incorporated 28 Research Drive Hampton, VA 23666	11. Contract or Grant No. NAS1-16303
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

NASA Langley Technical Monitor: Wayne Bryant

16. Abstract

This report documents the computer program PIFCGT. The FORTRAN program, PIFCGT, is a computer design aid for determining Proportional-Integral-Filter (PIF) control laws for aircraft autopilots implemented with a Command Generator Tracker (CGT). The program uses Linear-Quadratic-Regulator synthesis algorithms to determine feedback gains, and includes software to solve the feedforward matrix equation which is useful in determining the command generator tracker feedforward gains. The program accepts aerodynamic stability derivatives and computes the corresponding aerodynamic linear model. The nine autopilot modes that can be designed include four maneuver modes (ROLL SEL, PITCH SEL, HDG SEL, ALT SEL), four final approach models (APR GS, APR LOCI, APR LOCR, APR LOCP), and a BETA HOLD mode. The program has been compiled and executed on a CDC computer.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Optimal Control-Application Digital Flight Control System Software Design Package General Aviation Flight Control System	FEDD Distribution Subject Category 08

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	176	

Available: NASA's Industrial Applications Centers

End of Document